

Relatório do Trabalho 1 - Introdução à Inteligência Artificial

Ian Nery Bandeira - 17/0144739
André Carvalho Marques - 15/0005491

Resumo: Esse trabalho consiste na implementação de um algoritmo genético, com duas formas diferentes de *crossover*, com até 1000 gerações, para solucionar o problema do Caixeiro Viajante com 10 cidades. Diante da análise e síntese dos dados, nos foi mais proveitoso para encontrar o caminho ótimo ao utilizar do método de *crossover* com um ponto, em detrimento do *crossover* de dois pontos.

Palavras-chave: Inteligência Artificial; Caixeiro Viajante; Algoritmo Genético;

1 Introdução

Algoritmos Genéticos são um método de otimização baseado em mecânicas de genética e seleção naturais. Um algoritmo genético mimetiza o princípio da genética para constituir procedimentos de busca e otimização, ou seja, são utilizados para angariar dados que procuram ser próximos a solução ótima de problemas, em pouco tempo. Algoritmos genéticos normalmente funcionam da seguinte forma, com alguns passos variáveis:

Primeiro, são selecionados “cromossomos”, que são em nível de algoritmo candidatos para a solução ótima do problema. Então, é feita a evolução dos cromossomos, que consiste no processo de selecionar alguns ou todos os cromossomos e criar novos cromossomos baseados em mudanças por *crossover*, utilizando seleção por ciclos, ou por mutações. Depois, tanto esses cromossomos iniciais quanto os gerados pela evolução são avaliados baseado no critério necessário para determinar a otimização da resposta. Com essa resposta, os cromossomos são “podados”, ou seja, são eliminados os cromossomos com a menor avaliação, de preferência com igual quantidade a cromossomos criados, e o ciclo se reinicia.

No problema apresentado, uma derivação do conhecido problema do Caixeiro Viajante, é pedido para que seja feito a implementação de um algoritmo genético, com duas formas diferentes de *crossover*, com até 1000 gerações, para solucionar o problema com 10 cidades da América Latina, começando e terminando em Brasília (BSB).

Esse relatório é organizado como segue. Na seção 2, são expostos os materiais de base utilizados para o funcionamento do algoritmo, bem como os métodos empregados para gerar e sintetizar os dados apresentados na seção seguinte. Na seção 3 são apresentados os dados provenientes da execução do algoritmo com ambas as estratégias de *crossover*, devidamente sintetizados em histogramas. Na seção 4, serão analisados os dados apresentados na seção anterior, e na seção 5 teremos as considerações finais acerca do projeto.

2 Materiais e métodos

Como fundamento teórico, foram utilizados os slides e vídeo aulas do professor Díbio, bem como o link disponibilizado por ele:

<https://towardsdatascience.com/evolution-of-a-salesman-a-complete-genetic-algorithm-tutorial-for-python-6fe5d2b3ca35>.

Como ferramentas empregadas no trabalho, foram utilizadas para a execução do algoritmo genético as bibliotecas *random*, como artifício tanto para escolher os cromossomos quanto para escolha de que estratégia de *crossover* o algoritmo iria utilizar (50% de chance de utilizar a estratégia de *crossover* de um ponto, que chamaremos de estratégia 1, e 50% de chance de utilizar a estratégia de dois pontos, que chamaremos de estratégia 2). Também foi utilizado uma lista de tuplas *matriz_eucl*, que consiste de todas as distâncias presentes na matriz euclidiana de distâncias exposta no roteiro do trabalho, onde as tuplas são (*origem*, *destino*, *distância*); e uma lista *idades_list*, com seus elementos sendo cada uma das cidades presentes na matriz euclidiana.

Para coleta e síntese dos dados sensíveis ao que foi pedido no trabalho, foram utilizadas as bibliotecas *random*, para gerar números aleatórios para os arquivos de texto responsáveis por conter tanto a rota ótima decidida pelo algoritmo, quanto seu valor, em quilômetros; a biblioteca “*os*”, para receber o caminho onde o projeto está, para salvar os arquivos e imagens nas pastas correspondentes; e a biblioteca “*pyplot*”, proveniente da “*matplotlib*”, utilizada para criar os gráficos e histogramas expostos na próxima seção.

3 Resultados quadro, gráficos e figuras

Segue como se comportaram as distribuições da distância em função do número de iterações no algoritmo, com ambas estratégias de *crossover*, e em seguida os histogramas provenientes da execução do algoritmo 1350 vezes, sendo 647 utilizando a estratégia 1 e 703 utilizando a estratégia 2.

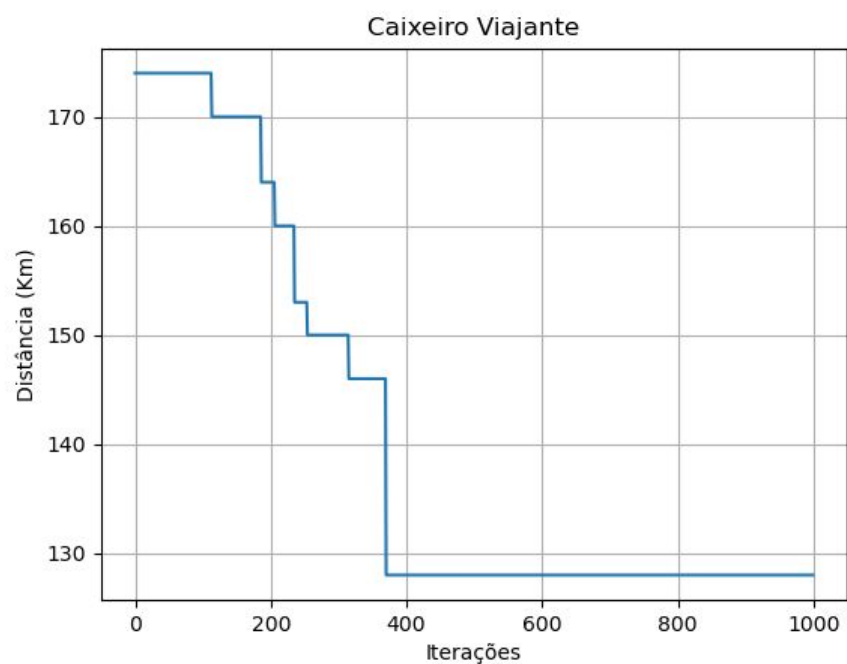


Gráfico 1 – Diagrama proveniente de uma das 647 execuções do algoritmo utilizando a estratégia 1

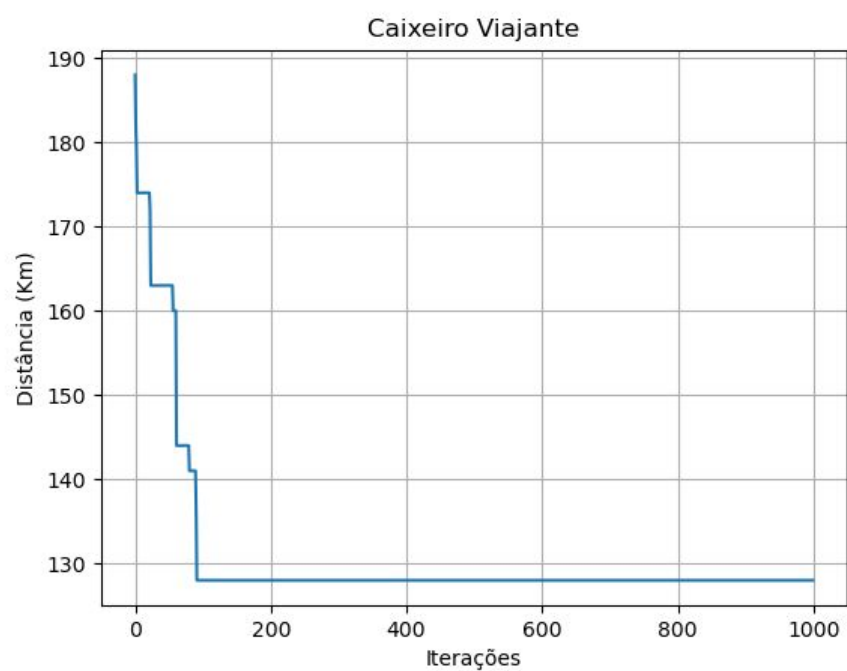


Gráfico 2 – Diagrama proveniente de uma das 703 execuções do algoritmo utilizando a estratégia 2

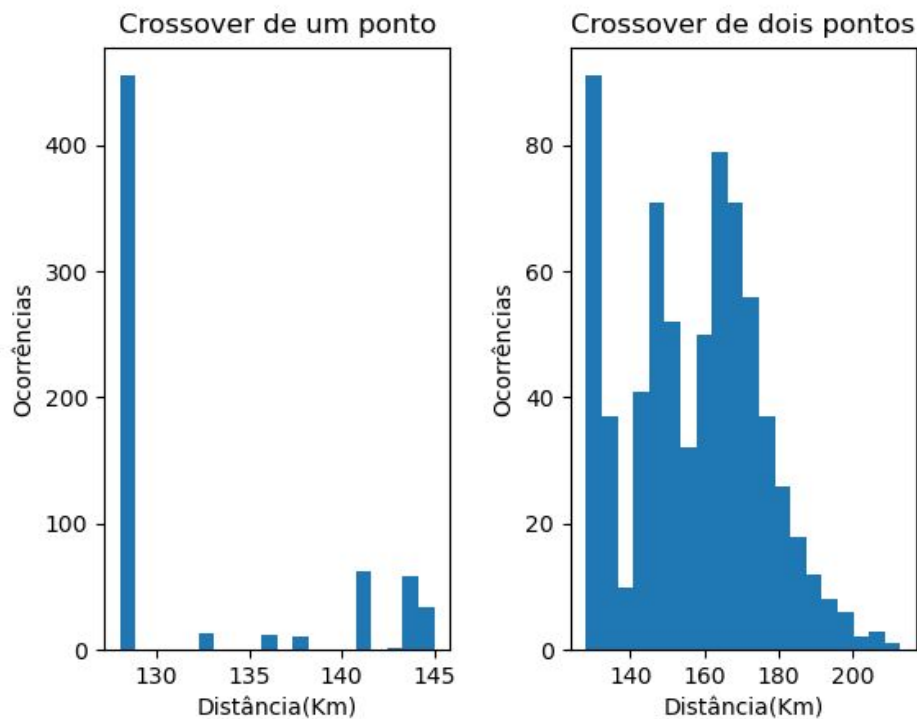


Gráfico 3 – Histograma proveniente de 1300 execuções do algoritmo, mostrando a ocorrência de cada distância em ambas as estratégias.

4 Análise de Resultados

O algoritmo gerou, utilizando ambas as estratégias de *crossover*, o mesmo menor caminho na maioria das ocorrências, que é o caminho:

BSB → BH → RJ → SP → PoA → BA → Sant → Lima → Bog → Carac → BSB

ou este ao contrário, começando em BSB e indo para Carac, e etc; e que ele possui distância total de 128 km, que é o menor valor observado no eixo de Distância(km) dos gráficos.

A observação latente que a análise do Gráfico 1 e do Gráfico 2 nos dão, é que existe uma disparidade entre a convergência de ambos os métodos, onde a convergência da estratégia 1 para o valor de distância ótimo é de aproximadamente 380 iterações, enquanto a convergência da estratégia 1 para a distância ótima leva aproximadamente 150 iterações; tendo em mente que ambos os gráficos foram escolhidos por apresentarem padrões similares entre si.

Ao analisar o histograma, porém, conseguimos perceber que há uma consistência muito maior no algoritmo que utiliza a estratégia 1 para encontrar o caminho ótimo, em detrimento do que utiliza a estratégia 2, apesar de em ambos os casos, o caminho ótimo ter sido escolhido na maior parte das ocorrências.

5 Considerações Finais/Conclusões

A partir dos dados apresentados e analisados, podemos afirmar que, apesar dos erros humanos e de randomicidade de ambas as estratégias, pois há randomicidade na escolha dos cromossomos, podemos afirmar que dada a quantidade de ocorrências, o algoritmo que utiliza a estratégia de *crossover* de um ponto se sobressai em consistência para escolha do melhor caminho, mas peca em desempenho; enquanto a estratégia de *crossover* de dois pontos se sobressai em desempenho, pois converge mais rápido, mas possui pior consistência quando comparada com a estratégia 1.

Referências Bibliográficas

HOLLAND, JOHN H. **Adaptation in Natural and Artificial Systems**, MIT Press, 1992.

ERIC STOLTZ **Evolution of a salesman: A complete genetic algorithm tutorial for Python**, 2018, disponível [Aqui](#).

JOHN HUNTER, DARREN DALE, ERIC FIRING, MICHAEL DROETTBOOM AND THE MATPLOTLIB DEVELOPMENT TEAM, **Matplotlib Documentation**, 2012 - 2020, disponível [Aqui](#).

PYTHON SOFTWARE FOUNDATION, **Python HOWTOs**, 2001-2020, disponível [Aqui](#).