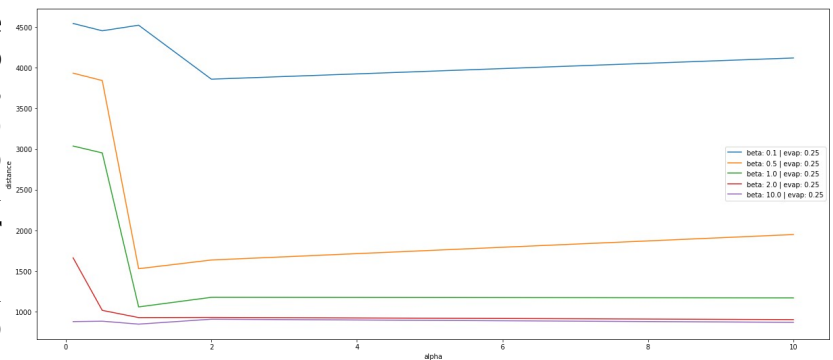


1. Trabalho Desenvolvido

Implementei uma solução modular para o problema em questão, que se resume a 3 classes em Python: uma para representar o **Traveling Salesman Problem** (suportando diferentes dimensões do problema e configurações aleatórias), outra para instanciar o algoritmo **Ant Colony Optimization (ACO)** (podendo ser aplicada a vários tipos de problemas, desde que estes implementem certas funções), e uma para instanciar um algoritmo que utiliza a heurística **Nearest Neighbour (NN)** (para servir como termo de comparação para averiguar a eficácia do ACO em relação às soluções mais tradicionais). Uma vez implementada a solução, fiz um **estudo do impacto dos diferentes parâmetros do ACO** nos seus resultados (em termos de **comprimento da solução** e número de **iterações necessárias** para o algoritmo convergir). Feito este estudo, selecionei os parâmetros que originaram os melhores resultados e apliquei-os a uma **instância de maiores dimensões** (200 nodos) do TSP, avaliando os resultados do ACO e **comparando-os** com os obtidos pela heurística Nearest Neighbour (em termos de comprimento da solução e tempo de execução), bem como avaliando o **perfil de convergência** do ACO neste último teste.

2. Resultados Obtidos

Fazendo variar a **taxa de evaporação** de feromonas no conjunto {0.25, 0.50, 0.75}, percebi que os melhores resultados ocorreram para o primeiro valor (e esse valor é também o que distancia mais o ACO da heurística NN, uma vez que privilegia o **fator feromona** nas tomadas de decisão). Fixando este parâmetro, estudei a variação do comprimento da solução para diferentes combinações dos parâmetros **alfa** e **beta**. As melhores soluções, neste problema, são originadas por valores altos no parâmetro beta. O parâmetro alfa é o que permite que este algoritmo combata os mínimos locais de visibilidade (e para os quais o NN não tem resposta), no entanto, valores altos neste parâmetro prejudicam, de um modo geral, a qualidade das soluções.

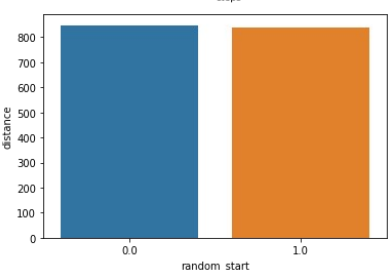
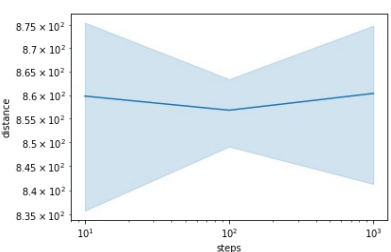
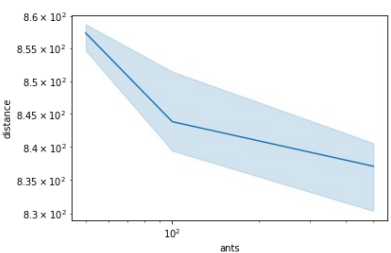


De seguida estudei a variação da qualidade da solução para diferentes **dimensões da colônia** e para diferentes **períodos de convergência** (número de iterações sem melhorias necessárias para parar o algoritmo). O **número de formigas beneficia largamente a exploração** e, por conseguinte, a qualidade das soluções encontradas. Por outro lado, o **período de convergência não tem um impacto tão significativo**.

Testei ainda a possibilidade de a exploração das formigas ser feita de maneira que metade da população das formigas partisse de um nodo aleatório, no entanto a diferença não foi significativa.

Finalmente, para testar o comportamento do ACO numa instância mais complexa do TSP (200 nodos) escolhi os parâmetros {taxa de evaporação: 0.25, alfa: 0.5, beta: 10, número de formigas: 250, período de convergência: 10 iterações}. Os resultados finais permitem-me tirar as seguintes principais **conclusões**:

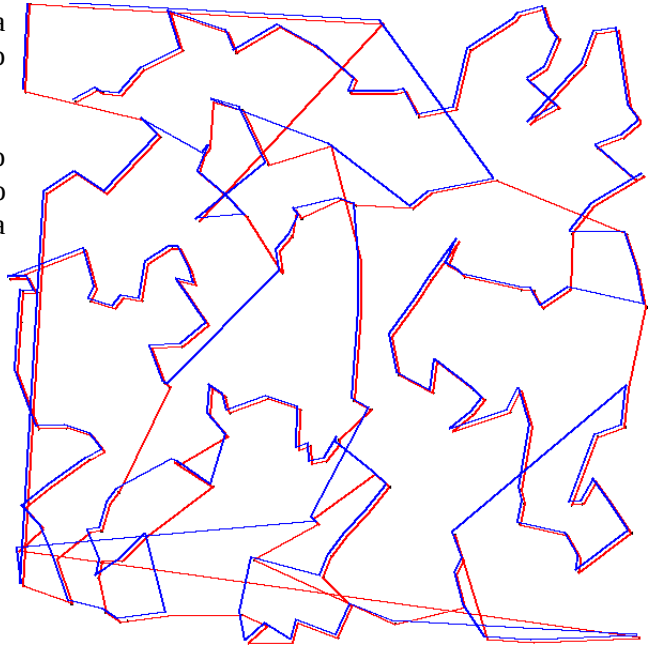
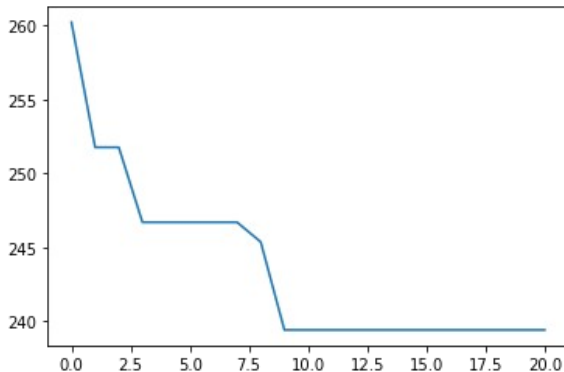
- A exploração introduzida pelo comportamento das formigas ajuda a ultrapassar mínimos locais e atingir soluções **ligeiramente melhores** do que as retornadas por heurísticas tradicionais **regidas pelo mesmo tipo de critérios** (neste caso o custo de casa passo, como critério de visibilidade no ACO e como critério único no NN).
- O custo desta ligeira melhoria na qualidade das soluções (em termos de tempo de execução) é **extremamente elevado**, e deverá apenas ser considerado quando existem os recursos disponíveis para tal e quando nos encontramos em contextos em que ligeiras melhorias tenham um valor significativo.



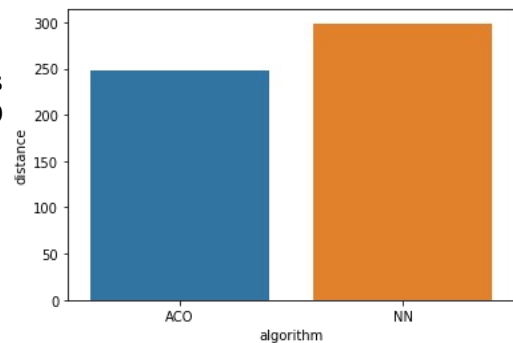
Abaixo encontram-se representações gráficas dos resultados finais.

À direita, os caminhos gerados pelo ACO (a vermelho) e pela heurística NN (a azul). Como podemos ver, as soluções são bastante próximas.

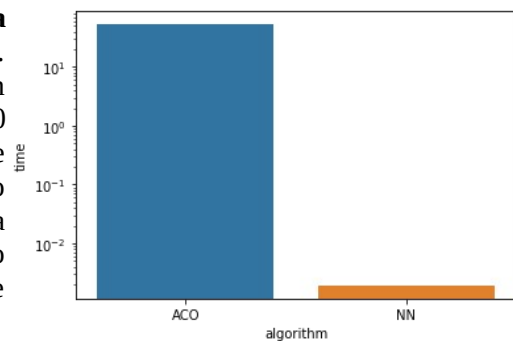
Abaixo, o comprimento da solução do ACO ao longo das iterações do algoritmo, demonstrando que este converge em menos de 10 iterações para esta configuração do TSP.



À direita, é feita a comparação entre a qualidade das soluções (comprimento do caminho escolhido) pelo ACO e pelo NN. O **ACO atingiu uma solução de qualidade superior.**



À direita, a comparação do tempo de execução (**escala logarítmica**) entre as duas abordagens à solução do problema. Naturalmente, a abordagem que utiliza a heurística NN tem um tempo de execução mais de **3 ordens de grandeza** (1000 vezes!) menor. Esta diferença é compreensível, uma vez que podemos encarar a heurística NN como um caso particular do ACO em que temos apenas uma formiga, apenas uma iteração, e não existe componente aleatória na escolha do próximo passo (apenas é tido em conta o fator de visibilidade). (à direita)



3. Desafios e Trabalho Futuro

O principal desafio que enfrentei foi encontrar uma forma de otimizar os parâmetros do algoritmo para configurações complexas do TSP, uma vez que os tempos de execução tornam impeditiva uma busca em rede (Grid Search) num espaço de parâmetros minimamente abrangente.

A solução que utilizei foi dividir os parâmetros em subconjuntos e proceder à optimização separada de cada subconjunto, no entanto esta abordagem não é ideal, pelo que uma possível melhoria a esta investigação passaria por utilizar ou implementar um método mais rigoroso e eficiente de avaliar o impacto de cada parâmetro do algoritmo na qualidade da solução e no número de iterações necessárias à convergência.

Outra melhoria que aumentaria bastante a eficácia e eficiência do ACO seria a implementação de diferentes meta-heurísticas para determinar o fator de visibilidade dos diferentes nodos, como por exemplo determinar o nodo “mais visível” a partir de uma seleção feita periodicamente dos K-nodos mais próximos de uma determinada localização ou região. A implementação de estratégias deste género, no entanto, torna o algoritmo cada vez menos genérico e mais especializado no problema em questão.