

Trabalho 01 - Problema do Caixeiro Viajante

Iago Silva [2022035881]

1 Problema

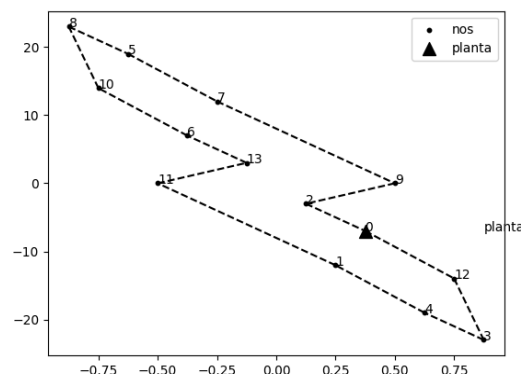
O Problema do Caixeiro Viajante é um dos problemas mais estudados em otimização combinatória e logística. Se trata da busca de determinar a rota mais curta que um viajante deve seguir para visitar um conjunto de cidades e retornar à cidade de origem, passando por cada cidade exatamente uma vez. Matematicamente, é um problema de minimização do comprimento total da rota, sujeito à restrição de visitar cada cidade uma única vez. Apesar de sua formulação simples, o problema é NP-difícil, o que significa que não se conhece um algoritmo eficiente que possa resolver todas as instâncias em tempo polinomial. A importância deste problema reside em sua aplicabilidade em diversas áreas, como logística, planejamento de rotas e até mesmo na genética.

2 Instâncias utilizadas

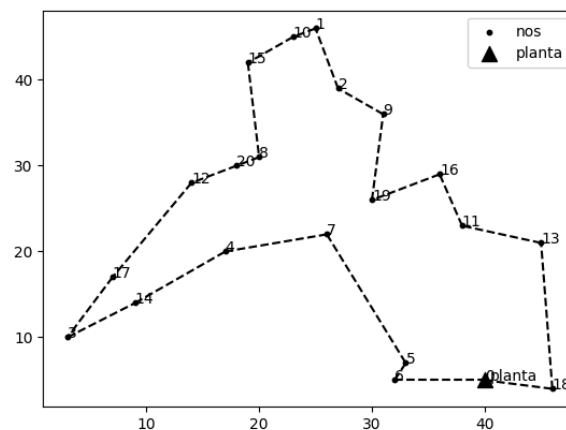
1. **Instância wc-instance-14** é um exemplo de problema do caixeiro viajante, porém com características diferentes da próxima instância berlin52. Nesta instância, o problema consiste em visitar 14 localidades específicas. Ela procura encontrar um caminho eficiente em um espaço de busca menor em comparação com instâncias maiores, como a berlin52. [N = 14]
2. **Instância Rivadália**, diz respeito a empresa de móveis Rivadália que possui uma fábrica e 20 clientes, os quais visita semanalmente para repor estoque. O caminhão que faz as entregas não tem restrição de capacidade, parte da fábrica, visita todos os clientes e retorna ao ponto original de partida, a fábrica. Essa rota deve ser definida de modo que a distância percorrida seja a menor possível. [N = 21]
3. **Instância berlin52** é um problema clássico usado para testar algoritmos de otimização para o Problema do Caixeiro Viajante. Nessa instância, o viajante precisa visitar todas as 52 localidades em Berlim e retornar ao ponto de partida, minimizando a distância total percorrida. [N = 52]

3 Discussão dos resultados

1. O algoritmo executa a wc-instance-14 de forma eficiente. A rota gerada tem um **custo total de 92.10**, o que indica que o algoritmo conseguiu encontrar uma rota com um custo baixo, considerando o problema. A rota começa e termina no nó 0, passando por todos os outros nós uma única vez, o que está de acordo com as regras do problema do caixeiro viajante. Isso sugere que o algoritmo está funcionando corretamente. Dado o tamanho da instância, o tempo de execução foi rápido, levando apenas 8 segundos. **ROTA: [0 12 3 4 1 11 13 6 10 8 5 7 9 2]**



2. Para a instância Rivadália, o algoritmo foi capaz de encontrar uma rota eficiente. A rota gerada possui um **custo total de 165.06**, indicando que o algoritmo conseguiu minimizar a distância percorrida, o que é crucial para uma empresa como a Rivadália, que precisa otimizar suas entregas semanais. A rota começa na fábrica (nó 0), visita todos os 20 clientes uma única vez e retorna à fábrica, conforme especificado no problema. Esse resultado sugere que o algoritmo está operando conforme esperado, conseguindo resolver o problema do caixeiro viajante de forma eficaz, considerando um tempo de execução também baixo, em torno de 13 segundos. **ROTA: [0 18 13 11 16 19 9 2 1 10 15 8 20 12 17 3 14 4 7 5 6]**



3. O algoritmo executa a instância berlin52 de forma eficiente. A rota gerada tem um **custo total de 7544.37**, o que indica que o algoritmo conseguiu encontrar uma rota com um custo bem mais alto que as duas instâncias anteriores. Comparei o custo de nossa execução com outros relatórios pela internet da instância e os resultados são iguais. A rota começa e termina no nó 0, passando por todos os outros nós uma única vez, o que está de acordo com as regras do problema do caixeiro viajante. Isso sugere que o algoritmo está funcionando corretamente. Dado o tamanho da instância, o tempo de execução foi um pouco maior, chegando a aproximadamente 3 minutos..**ROTA [0 48 31 44 18 40 7 8 9 42 32 50 10 51 13 12 46 25 26 27 11 24 3 5 14 4 23 47 37 36 39 38 35 34 33 43 45 15 28 49 19 22 29 1 6 41 20 16 2 17 30 21].**

