



Universidade Federal Rural de Pernambuco
Departamento de Estatística e Informática
Bacharelado em Sistemas de Informação

FlyFood

Lucas Gabriel Carvalho dos Ramos

Recife

Fevereiro de 2024

1. Objetivos

O propósito do projeto consiste na criação e execução de um algoritmo eficaz que visa aprimorar as rotas de entrega de um drone, assegurando a realização de todas as tarefas no tempo de vida da bateria, por meio da identificação do trajeto mais breve para maximizar a eficiência operacional.

Objetivos específicos

Força Bruta

- Extração das Coordenadas da Matriz: O código lê uma matriz de um arquivo .txt e extrai as coordenadas dos pontos de entrega.
- Cálculo da Distância entre Pontos: O código calcula a distância entre dois pontos em um plano cartesiano para determinar as distâncias entre os pontos de entrega.
- Geração de Todas as Permutações de Pontos de Entrega: O código gera todas as possíveis ordens de entrega usando a função `permutations` do módulo `itertools`.
- Cálculo do Custo Total: O código calcula o custo total de uma determinada ordem de entrega, considerando as distâncias entre os pontos na ordem de entrega.
- Determinação da Ordem de Entrega Usando Força Bruta: O código encontra a ordem de entrega ótima gerando todas as permutações possíveis dos pontos de entrega e selecionando a que tem o menor custo total.
- Remoção de 'R' Duplicados Consecutivos: O código remove 'R' duplicados consecutivos na ordem final para otimizar a rota de entrega.
- Impressão da Melhor Ordem de Entrega e do Menor Custo: O código imprime a melhor ordem de entrega e o menor custo.

Vizinho mais Próximo

- Extração das Coordenadas da Matriz: O código é responsável por ler uma matriz de um arquivo .txt e extrair as coordenadas dos pontos de entrega, garantindo uma representação clara da origem dos dados.
- Cálculo da Distância entre Pontos: Ele calcula a distância entre dois pontos em um plano cartesiano para determinar as distâncias entre os pontos de entrega, garantindo precisão nas medidas.
- Determinação do Próximo Ponto: O programa encontra o próximo ponto a ser visitado com base na menor distância do ponto atual, utilizando o algoritmo do vizinho mais próximo para otimizar o percurso de entrega.
- Cálculo do Custo Total: Ele calcula o custo total de uma determinada ordem de entrega, considerando as distâncias entre os pontos na ordem de entrega, proporcionando uma estimativa precisa do custo associado à entrega.
- Determinação da Ordem de Entrega Usando o Vizinho Mais Próximo: Por meio de um loop, o código encontra a ordem de entrega ótima utilizando o algoritmo do vizinho mais próximo, garantindo eficiência e redução do tempo de entrega.

2. Procedimento

Força Bruta

O programa começa na função `"main"`, onde o arquivo com a matriz de pontos é lido. Uma lista dos pontos de entrega é gerada a partir das coordenadas. Todas as possíveis ordens de entrega são geradas com a biblioteca `"itertools"`, e o custo total é calculado para cada uma. A ordem com o menor custo total é identificada. Após ajustes para legibilidade, os resultados finais são exibidos, incluindo a melhor ordem de entrega e o custo total mais baixo.

Vizinho Mais Próximo

O programa começa na função "main", onde o arquivo de matriz de pontos é lido. Utilizando o algoritmo do vizinho mais próximo, é encontrada a ordem ótima de entrega. O custo total é calculado considerando as distâncias entre os pontos. Ao iterar sobre os pontos na função principal, a ordem é armazenada e o custo final é calculado. Por fim, a ordem de entrega e o custo total são exibidos na tela.

Configurações da máquina: Processador: Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz; Ram: 8 Gb; Placa de vídeo: GTX 1650 Gb

preparação da máquina: O notebook foi desligado antes de todos os testes para impedir que o tempo de processamento tivesse interferência de programas externos e para cada mapa foram testadas 3 execuções.

Teste 1:

0	0	0	0	D
0	A	0	0	0
0	0	0	0	C
R	0	B	0	0

Matriz 4x5

Testes 2:

0	0	0	0	D	0	0
0	A	0	0	0	0	J
0	0	0	0	C	0	0
R	0	B	0	0	L	0
0	0	0	0	E	0	0
0	F	0	0	0	0	0

matriz 6x7

Teste 1 - Vizinho mais próximo (E = EXECUÇÃO) Custo total: 16

Tempo de execução em segundos: E1 = 0.0059; E2 = 0.0; E3 = 0.0012.

Teste 1 - Força Bruta (E = EXECUÇÃO) Custo total: 14

Tempo de execução em segundos: E1 = 0.0009; E2 = 0.001; E3 = 0.001.

Teste 2 - Vizinho mais próximo (E = EXECUÇÃO) Custo total: 38

Tempo de execução em segundos: E1 = 0.0010; E2 = 0.0019; E3 = 0.0019.

Teste 2 - Força Bruta (E = EXECUÇÃO) Custo total: 28

Tempo de execução em segundos: E1 = 11.74; E2 = 11.49; E3 = 11.46.

O algoritmo do vizinho mais próximo é mais rápido computacionalmente, calculando apenas as distâncias entre pontos adjacentes. No entanto, pode não produzir a solução ótima, como indicado por custos totais mais altos em testes. Por outro lado, o algoritmo de força bruta considera todas as permutações possíveis, garantindo a solução ótima, mas é significativamente mais lento, especialmente para matrizes maiores. A escolha entre os dois depende das necessidades específicas do problema: se a precisão for crucial e o tempo não for uma preocupação, o algoritmo de força bruta é preferível; se a eficiência de tempo for mais importante e uma solução aproximada for aceitável, o algoritmo do vizinho mais próximo é a melhor opção.

Referências Bibliográficas

<https://programadorviking.com.br/abs-python-como-obter-o-valor-absoluto-guia-completo/>

<https://www.freecodecamp.org/news/with-open-in-python-with-statement-syntax-example/>

<https://pt.stackoverflow.com/questions/289538/o-que-significa-floatnan-e-floatinf-no-python>