
PROJETO 1

Eduarda Grazielle de Paiva
cc22125@g.unicamp.br

Eloisa Paixão de Oliveira
cc22127@g.unicamp.br

1 Introdução

O problema do caixeiro-viajante (PCV) é um desafio que busca encontrar a rota mais curta para visitar uma série de cidades, retornando à cidade de origem e passando por cada cidade apenas uma vez. Embora existam algoritmos de aproximação eficazes para o PCV, ele continua a atrair a aplicação de novos métodos, como os algoritmos evolutivos. Isso se deve à sua fácil compreensão e relevância para problemas do mundo real. Há diversos conjuntos de dados disponíveis na literatura para o PCV, permitindo comparações entre resultados, embora o ótimo global ainda não seja definitivamente conhecido. Esse problema representa uma ampla classe de desafios para os quais não há algoritmos polinomiais determinísticos.[6][2]

O algoritmo desenvolvido para este projeto visa resolver o desafio dos múltiplos caixeiros viajantes (mTSP). Ele segue o mesmo princípio do problema original, porém diferencia-se ao envolver mais de um caixeiro, cada um com um conjunto específico de cidades para visitar, mantendo a regra de que nenhuma cidade pode ser visitada mais de uma vez e que cada cidade só pode ser percorrida por um único caixeiro.[2][5]

2 Heurística construtiva

A heurística usada para solucionar esse problema é a Heurística do Centroide. Ela é um método utilizado para resolver o problema do caixeiro-viajante (PCV) de forma aproximada. Neste método, o objetivo é encontrar um centroide, ou seja, um ponto médio, que represente uma posição central em relação às cidades que precisam ser visitadas.[4][6][3]

Iniciando com a escolha aleatória de uma cidade como ponto de partida, ou por critérios específicos, como a proximidade ao centro geométrico das coordenadas das cidades, depois o processo segue com o cálculo do centroide das cidades restantes. Esse centroide é uma média ponderada das coordenadas das cidades não visitadas. [4][1][3]

A partir daí, selecionamos a cidade mais próxima desse ponto para ser visitada, podendo utilizar diferentes heurísticas de seleção, como a distância mínima ou a minimização do tempo de viagem.[4][1][3]

Esse processo se repete até que todas as cidades tenham sido visitadas, recalculando o centroide a cada iteração para considerar as cidades restantes. Após visitar todas as cidades, o caixeiro-viajante retorna à cidade inicial para completar o circuito, formando assim um ciclo fechado onde cada cidade é percorrida uma única vez.[4][1][3]

É importante destacar que a heurística do Centroide oferece uma solução aproximada para o problema do caixeiro-viajante, não garantindo a solução ótima. No entanto, é uma abordagem eficaz e de implementação relativamente simples, adequada para situações práticas onde a busca pela solução perfeita é impraticável devido à complexidade computacional envolvida.[4][1][3]

3 Experimentos computacionais

Os experimentos foram feitos em um computador macOS Sonoma com um chip Apple M1, 16GB de memória RAM. O projeto foi implementado em Python 3.12.4 e resolvido com Visual Studio Code 2024.

3.1 Instâncias de testes

O conjunto de testes consiste em 9 instâncias, variando o número de cidades em 13, 17, 19, 31, 47, 59, 71, 83 e 91 e o número de caixeiros em 1, 3 e 5.

3.2 Resultados

A Tabela 1 a seguir resume os resultados obtidos para cada uma das instâncias. As colunas mostram em ordem o valor das instâncias, número de cidades (n), número de caixeiros (m), quantidade máxima que cada caixeiro pode visitar (k), valor da solução ótima para cada instância ($f(x)$) disponibilizada pelo professor e os resultados obtidos pelos testes do projeto (r).

| Instância | n | m | k | $f(x)$ | r |
|-------------|-----|-----|-----|--------|-------|
| mTSP-n13-m1 | 13 | 1 | 13 | 3071 | 4148 |
| mTSP-n17-m1 | 17 | 1 | 17 | 3948 | 5766 |
| mTSP-n19-m1 | 19 | 1 | 19 | 4218 | 7614 |
| mTSP-n31-m3 | 31 | 3 | 11 | 5841 | 8319 |
| mTSP-n47-m3 | 47 | 3 | 16 | 6477 | 12346 |
| mTSP-n59-m3 | 59 | 3 | 20 | 6786 | 12078 |
| mTSP-n71-m5 | 71 | 5 | 15 | 8618 | 16762 |
| mTSP-n83-m5 | 83 | 5 | 17 | 9246 | 16911 |
| mTSP-n91-m5 | 91 | 5 | 19 | 9586 | 19082 |

Tabela 1 : Instâncias

Ao analisar os resultados das soluções obtidas, observa-se uma disparidade em relação às soluções ótimas disponibilizadas. Enquanto algumas soluções aumentaram em 20, outras apresentaram um aumento mais significativo de 50, porém nenhuma delas ultrapassou os limites máximos das soluções ótimas. Essas disparidades eram esperadas, como já foi explicado anteriormente, uma vez que a heurística utilizada não é a mais adequada para a resolução desse problema específico. No entanto, sua eficácia é comprovada pela simplicidade de sua implementação.

4 Referências

Referências

- [1] CHATGPT.
- [2] DA CUNHA, C. B., DE OLIVEIRA BONASSER, U., AND ABRAHÃO, F. T. M. Experimentos computacionais com heurísticas de melhorias para o problema do caixeiro viajante. In *XVI Congresso da Anpet* (2002).
- [3] DA SILVA, B. J. B., GONÇALVES, L. B., AND ARROYO, J. E. C. Heurística ils para minimização do consumo de energia em redes de sensores sem fio.
- [4] DE CARVALHO, R., SALDANHA, R. R., AND MARTINS, A. X. Heurística aplicada ao problema de agrupamento centrado capacitado.
- [5] OLIVEIRA, A. F. M. D. A. Extensões do problema do caixeiro viajante. Master's thesis, 2015.
- [6] WIKIPÉDIA. Problema do caixeiro-viajante.