

# Múltiplo Caixeiro Viajante

Victor Hugo Grabowski Beltramini<sup>1</sup>,  
André Elias Zanella<sup>2</sup>

<sup>1</sup>Universidade do Estado de Santa Catarina (UDESC)  
Ibirama, SC, Brasil

vhbeltramini@gmail.com, andreellias19@gmail.com

**Abstract.** *This article addresses a complex optimization problem, which consists of finding the best route for a network of brand representatives to travel through cities, starting from the same city and ending in the same city so that all cities be covered in the shortest possible distance. This article aims to solve this optimization problem by using a set of metaheuristics that contemplate constructive and modification strategies. Finally, a study is carried out experimental based on the results obtained.*

**Resumo.** *Este artigo contempla um problema de otimização complexo, que consiste em encontrar a melhor rota para um grupo de  $N$  caixeiros percorrerem  $C$  cidades, partindo da mesma cidade e terminando na mesma cidade de forma que todas as cidades sejam percorridas na menor distância possível, este problema recebe o nome de caixeiros viajantes múltiplos. O presente artigo tem como objetivo resolver esse problema de otimização, por meio da utilização de um conjunto de metaheurísticas que contemplam estratégias construtivas e de modificação. Por fim é realizado um estudo experimental com base nos resultados obtidos.*

## 1. Introdução

O problema do caixeiro viajante múltiplo (mTSP) é uma versão mais complexa do problema clássico de otimização que é o caixeiro viajante, este problema se baseia em definir uma rota que tenha o menor custo para que um caixeiro que percorra um número  $x$  de cidades, e na variação abordada nesse trabalho podemos ter  $n$  caixeiros ao invés de um como no problema inicial. Esse problema pode ser aplicado em diversos casos práticos, como planejamento de caminho tridimensional, planejamento de caminhos para robôs e drones ou serviço de transporte de contêineres.

O objetivo principal do trabalho é criar metaheurísticas que possam de forma efetiva diminuir o custo da função objetiva até um nível aceitável de satisfação. Para isso foram criadas 2 heurísticas, 2 buscas locais e 2 GRASP. Ao final é feita uma análise de todos os dados obtidos para se obter uma melhor compreensão da proporção de melhora obtida.

## 2. Problema

Este problema contempla a dificuldade de um grupo de indivíduos precisarem passar por um grupo de cidades, de modo que todas as cidades sejam percorridas por no máximo um indivíduo e que todos iniciem e terminem na mesma cidade. Com isso, no final deseja-se

obter a solução em que a soma das distâncias da rota de cada indivíduo seja a menor possível.

Para otimizar esse problema foi utilizada uma representação em grafo, onde cada vértice é uma cidade e as arestas representam a distância entre as cidades. Assim ao somar as distâncias do caminho de cada representante obteremos a função objetivo. Com isso obtemos os dados onde  $i$  é o número de cidades,  $t$  é o número de caixeiros e  $D_{ij}$  é a matriz de distâncias entre cada cidade. A seguir temos o modelo matemático do problema:

$$\begin{aligned}
& \text{minimize} \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\
& \text{subject to} \quad \sum_{j=1}^n x_{ij} = 1, i = 1, \dots, n \\
& \quad \quad \quad \sum_{i=1}^n x_{ij} = 1, j = 1, \dots, n \\
& \quad \quad \quad u_i - u_j + nx_{ij} \leq n - 1, i, j = 2, \dots, n \\
& \quad \quad \quad x_{ij} \in \{0, 1\},
\end{aligned}$$

**Figura 1. Modelo matemático**

### 3. Algoritmos

Inicialmente foram feitas 2 heurísticas construtivas, onde a primeira utiliza o critério de seleção como a menor soma encontrada para a distância entre a posição atual de cada caixeiro e a cidade  $c$ . Após achar a melhor soma (cidade) é então escolhido o caixeiro com a menor distância para essa cidade e que incluirá essa cidade na rota.

Para a segunda heurística foi utilizado o critério de seleção como a menor distância encontrada entre os caixeiros em suas posições atuais, incluindo assim essa cidade para esse caixeiro. Além disso para cada heurística foram feitas 2 implementações, sendo a primeira a gulosa, onde só é pego a menor distância encontrada e a segunda uma semi-gulosa do tipo alfa-gulosa, onde é pego um valor aleatório entre os 50% melhores valores obtidos segundo cada respectivo critério.

Também foram feitas 2 buscas locais, as duas buscas locais tiveram como critério de aceitação somente soluções que melhorassem a solução inicial, para que essa solução melhorada pudesse ser atingida foram implementadas duas maneiras de perturbações diferentes, uma delas sendo um shift entre rotas e a segunda aplicando um shift intra-rotas, após cada uma dessas melhoras era definida esta nova solução como sendo melhor, e portanto assim o algoritmo seguia com essa nova solução a fim de encontrar uma solução ainda melhor.

Por fim foram desenvolvidos e aplicados 4 algoritmos GRASP (Greedy Randomized Adaptive Search Procedure), combinando as duas heurísticas construtivas na sua implementação semi-gulosa com as duas buscas locais. A Heurística foi utilizada para a solução inicial e depois aplicado a busca local por uma quantidade  $x$  de vezes de forma que se obtenha uma melhor solução do que usando cada um de forma individual.

No implementações relacionadas ao GRASP o sistema se comportava da seguinte maneira, a partir de uma solução inicial advinda de uma heurísticas construtiva semi-gulosa era aplicada uma busca local afim de melhorar essa solução inicial e então era definida está como a melhor solução até o momento, após esta etapa inicial o sistema iterava x vezes realizando uma nova heurísticas construtiva semi-gulosa e novamente uma nova busca local a fim de se tentar encontrar uma solução melhor, caso não encontrava o sistema continuava com a melhor solução até o momento, e ao final das x iterações era retornada a melhor solução encontrada dentre as iterações.

#### 4. Resultados

Segue abaixo a tabela com os resultados obtidos, onde na figura 2 contém as duas heurísticas desenvolvids com seus respectivos resultados otimizados para cada instância. Na figura 3 é constado as 2 buscas locais e os 4 GRASPs implementados, cada um com o seu resltado para as instâncias utilizadas. Ao final temos uma coluna com o GRASP desenvolvido por [Davi Juliano Ferreira Alvesa 2022] (figura 3), que foi usado como referência para a comparação de resultados obtidos. Para cada metaheurística foi pego o resultado com a melhor otimização obtida ( $S^*$ ) e a média entre 3 resultados obtidos (S). As instâncias utilizadas durante a implementação e teste fora obtidas de [He and Hao 2022], onde as instâncias variam de 51 á 200 cidade e de 3 ou 5 caixeiros.

	H1(menor soma)	H2(menorDist)	H
Instância	S	S	S**
mtsp51_3	617.16	584,09	446,0
mtsp100_3	31,906.30	27,039.73	21797,6
mtsp150_3	52,615.44	46,390.92	23174,9
mtsp51_5	751.23	615.84	474,7
mtsp100_5	36,822.77	27,039.73	8223,9
mtsp150_5	63,249.03	50,183.94	39066,5
kroa200_3	44,662.68	40,362.70	29539,5

Figura 2. Heurísticas

Instância	LS1(Intra)		LS2(Inter)		GRASP1		GRASP2		GRASP3		GRASP4		GRASP	GRASP(Artigo ref)
	S	S*	S	S*	S	S*	S	S*	S	S*	S	S*	S**	
mtsp51_3	793.20	768.00	707.02	643.56	457.3	657.33	894.48	890.58	527.68	502.17	664.64	658.36	446.0	448.9
mtsp100_3	49,794.05	46,216.94	58,185.41	55,216.20	22451.7	46,808.00	84,356.89	81738.37	34,127.90	32,949.87	50,282.70	48773.67	21797.6	8042.2
mtsp150_3	75,665.12	74,521.99	101,117.43	96,951.18	38452.6	67,454.84	144,708.30	143002.05	54,108.57	53,329.07	90,553.54	90298.94	23174.9	38582.0
mtsp51_5	911.58	885.58	668.61	619.02	484.6	844.66	922.44	876.6	558.68	546.16	614.87	613.54	474.7	474.8
mtsp100_5	64,097.69	62,080.44	51,561.72	48,256.83	23797.6	59,165.48	92,773.80	91325.29	35,495.91	36,113.66	76743.41	76,512.64	23174.9	8397.1
mtsp150_5	95,543.56	87,666.90	85,614.33	81,047.61	39731.9	83,204.29	149,057.04	143898.82	57,654.81	57,568.72	46146.585	45,165.07	39066.5	6785.8
kroa200_3	72,088.52	67,296.10	286,687.62	99,846.42	30602.1	64,912.57	188,132.61	184890.35	51369.2	50323.39	96988.495	95034.95	30134.5	30523.6

Figura 3. Buscas locais e GRASPs

Como visto nos resultados, ainda há uma difença considerável entre o melhor resultado encontrado a partir da nossa implementação e a melhor solução já encontrada, porém tal diferença pode ser diminuída consideravelmente a partir uma implementação mais otimizada da função responsável pelo cálculo da função objetiva que avalia a cada iteração se houve ou não uma melhoria na solução.

Como solução usada atualmente não está fazendo essa avaliação da função objetiva de maneira mais otimizada possível a mesma acaba custando um tempo considerável

considerando o tempo total de execução para a definição de cada solução, isso acabou fazendo com que fosse diminuída o número de iterações até se encontrar a melhor solução a fim de que o processo não levasse muito tempo para entregar uma solução melhor.

## **5. Conclusão**

Com base nos resultados obtidos ao executar cada meta-heurística podemos perceber que a otimização de fato ocorreu, todas as instâncias apresentaram uma melhoria considerável. Único porém no entanto é que quando olhando exclusivamente as buscas locais e as heurísticas construtivas ainda possivelmente há melhorias que podem ser feitas para que se chegue mais próximas as melhores soluções já encontradas até o momento.

Portanto pode se concluir que as implementações atingiram sua proposta inicial que era de encontrar uma solução para o problema e ainda posteriormente foram capazes de melhorar solução uma inicial semi-gulosa que acaba trazendo um pouco de aleatoriedade para a solução mas ainda permitindo encontrar uma solução boa a fim de otimizar ao máximo possível a rota dos caixeiros para que fosse percorrida com a menor distância possível.

## **Referências**

- Davi Juliano Ferreira Alves, Bruno Avelino, V. C. N. d. F. E. A. A. G. M. V. G. M. B. S. V. A. A. C. (2022). Estudo comparativo de metaheurísticas aplicadas ao problema do caixeiro viajante múltiplo. *SBPO 2022*.
- He, P. and Hao, J.-K. (2022). Hybrid search with neighborhood reduction for the multiple traveling salesman problem. *LERIA, Université d'Angers*.