

**Identificação da Proposta:**  
**EDITAL PROPQ 002/2022 INICIAÇÃO CIENTÍFICA SEM**  
**REMUNERAÇÃO (ICTSR) E BOLSISTAS FINANCIADOS**  
**POR OUTRAS FONTES (ICTBOF): FLUXO CONTÍNUO**

**Heurísticas e Meta-Heurísticas Evolutivas**  
**para o TSP com Drone**

**07/2022**

# Heurísticas e Meta-Heurísticas Evolutivas para o TSP com Drone

**Candidato:** Lorenzo Correia Maia

**Orientador:** Mário César San Felice

**Coorientador:** Pedro Hokama

Departamento de Computação - UFSCar

São Carlos, 07/2022

## Resumo

O famoso problema do Caixeiro Viajante (TSP, do inglês *Traveling Salesman Problem*) fala sobre um vendedor que sai de sua cidade de origem e tem que visitar outras cidades, através de estradas que as liguem. Um dos pontos chave é que ele deve passar por cada uma das cidades apenas uma vez e voltar para a origem. Tendo como base o TSP, podemos aplica-lo a diversos cenários, sendo um deles o roteamento de veículos. Em geral, esse problema envolve decisões que minimizam o custo do uso de veículos para transportar produtos e atender a demandas geograficamente dispersas.

Mais recentemente drones tem sido usados em conjunto com meios convencionais de entrega. Nesse contexto surge o TSP com Drone (TSP-D) em que o caminhão de entregas pode utilizar um drone para visitar um cliente enquanto o caminhão segue sua rota. Este projeto visa estudar, projetar, implementar e testar algoritmos heurísticos e meta-heurísticos, com foco no BRKGA, para o TSP-D.

Como projeto de Iniciação Científica, esse projeto também tem por finalidade introduzir o candidato na área de pesquisa científica e auxiliar na sua formação em Ciência da Computação.

## 1 Introdução e Justificativa

Problemas de roteamento de veículos envolvem a construção de rotas visando minimizar custos de entrega de produtos ou prestação de serviços para os clientes. Em geral, problemas dessa família são computacionalmente desafiadores, a maioria sendo NP-difícil. Além disso, esses problemas são bastante relevantes na prática, pois modelam diversos cenários envolvendo problemas de logística, que impactam custos e disponibilidade de produtos e serviços, bem como a facilidade de deslocamento de pessoas. Um problema fundamental dessa área é o clássico problema do Caixeiro Viajante (TSP, do inglês *Traveling Salesman Problem*), que envolve a construção de uma única rota de custo mínimo que visita cada cliente exatamente uma vez.

Com o advento dos drones surgiu uma nova gama de problemas de roteamento com drones [6, 8], que tratam de entregas híbridas nas quais veículos convencionais,

como caminhões, carregam drones e podem decidir se cada cliente será atendido de forma convencional ou se irá enviar o drone para realizar a entrega no mesmo. Nessa família de problemas existem várias configurações envolvendo, por exemplo, um caminhão e um drone [1, 5], um caminhão com vários drones [10], e até múltiplos caminhões cada um carregando diversos drones [7]. Da configuração mais básica, que envolve um caminhão e um drone, destacamos o TSP com Drone (TSP-D). Assim como o TSP é central dentre os problemas de roteamento de veículos, o TSP-D é fundamental dentre os problemas de roteamento com drones. Isso porque, ao mesmo tempo que ele é mais simples por envolver apenas um caminhão e um drone, ele captura características centrais das entregas híbridas, como a possibilidade de bifurcação da rota.

Explicando melhor o TSP-D a ideia é que o veículo inicie sua rota, e em um determinado ponto ele possa soltar o drone com a carga correspondente à entrega dos produtos de um cliente, enquanto o veículo continua seu caminho, até que, após o drone realizar a tarefa, ele retornará para um ponto a frente na rota para que seja recolhido. Desta forma, essa ação se repetirá até que todos os produtos sejam entregues e o veículo volte a sede da empresa em que começou essa jornada. Mais precisamente, uma solução será composta de uma sequência  $TD$  de pontos atendidos pelo veículo, e um conjunto  $DD$  com os pontos atendidos pelo drone. No conjunto  $DD$  cada elemento é uma tripla  $\langle i, j, k \rangle$ , sendo  $i$  o ponto de lançamento,  $j$  o ponto de entrega e  $k$  o ponto de encontro com o veículo. A Figura 1 ilustrar uma rota do TSP-D.

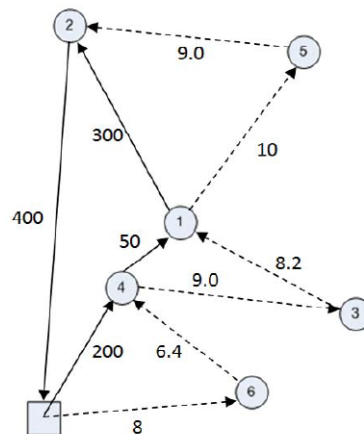


Figura 1: Solução para um TSP-D. Imagem retirada do artigo [5].

Entre as técnicas de projeto a serem estudadas nessa Iniciação Científica estão algoritmos gulosos, heurísticas de busca local [2], e a meta-heurística *Biased Random-Key Genetic Algorithm* (BRKGA) [4]. Ao longo deste projeto, alguns dos algoritmos estudados ou projetados serão implementados e testados nas linguagens C/C++ ou Python. Este projeto também visa apoiar a formação do candidato, que atualmente

curso Ciência da Computação na Universidade Federal de São Carlos (UFSCar). Como sua primeira iniciação científica, o projeto irá apresentá-lo ao trabalho de pesquisa e aprofundar seu conhecimento nas áreas de otimização combinatória, grafos e projeto e análise de algoritmos.

## **2 Abordagens de Pesquisa**

Em seguida, são apresentados conceitos e técnicas de projeto de algoritmos relevantes para este trabalho de pesquisa.

### **2.1 Otimização Combinatória**

A otimização combinatória é uma área da teoria da computação que trata da tomada de decisões com uma grande quantidade de informação, com o objetivo de obter o resultado ideal para a situação, seja maximização de lucros ou minimização de custos. Geralmente, um problema nesta área possui um conjunto de restrições que definem se uma solução é válida e uma função objetivo que determina o valor de cada solução. Devido ao alcance desta definição, esta área inclui vários problemas, como roteamento de veículos, localização de instalações, construção de redes e entre outras. Portanto, seus resultados encontram aplicações em diversas áreas, tanto dentro quanto fora da computação.

Um detalhe importante na área é que a maioria dos problemas de otimização combinatória são NP-difíceis. Portanto, a menos que  $P=NP$ , não existem algoritmos eficientes para resolver estes problemas de forma ótima para qualquer instância, sendo que consideramos eficientes algoritmos cujo tempo de execução é limitado por um polinômio no tamanho da instância.

Diferentes abordagens evitam essa dificuldade dispensando uma ou mais dessas restrições. Como estamos interessados em resolver instâncias grandes e gerais dos problemas, nossas abordagens são heurísticas e meta-heurísticas [3, 9] que envolvem vasculhar o espaço de busca a fim de encontrar boas soluções num tempo razoável, ainda que sem garantia de encontrar uma solução ótima.

### **2.2 Heurísticas**

Heurística para a computação, é uma técnica que tem por objetivo resolver problemas de forma eficiente. No entanto, em geral, algoritmos heurísticos produzem soluções sem dar garantias sobre a qualidade destas. Algumas técnicas de projeto de algoritmos que costumam produzir boas heurísticas estão apresentadas a seguir:

**Algoritmos gulosos.** Um algoritmo guloso ou míope é uma técnica que almeja encontrar o melhor caso global, utilizando de decisões que localmente são ótimas em cada instância do problema. Algoritmos clássicos em grafos como Kruskal e Dijkstra fazem uso dessa técnica e tem soluções ótimas. Entretanto, em geral algoritmos gulosos não garantem soluções ótimas, mas muitas delas são boas na prática. Algoritmos gulosos costumam ser usados para encontrar soluções viáveis para problemas NP-difíceis. Chamamos estes de heurísticas construtivas.

**Busca local.** Esses algoritmos, assim como os algoritmos gulosos, se baseiam em escolhas que obtêm melhorias locais sem garantia de atingir um ótimo global. Eles se diferenciam por começar com uma solução viável e, iterativamente, fazerem mudanças pequenas nesta solução, de modo a sempre transformá-la numa solução um pouco melhor. Para entender o funcionamento da busca local, precisamos entender o conceito de vizinhança entre soluções. Considere um grafo em que cada vértice é uma solução do problema e existe uma aresta entre dois vértices se as soluções correspondentes atendem a algum critério de similaridade. Entende-se por vizinhança o conjunto de vértices adjacentes ao vértice que está sendo analisado. Na busca local migramos iterativamente de uma solução para alguma solução vizinha que melhore o valor da função objetivo. Como estas melhorias podem ser muito pequenas, estes algoritmos não necessariamente terminam em um número polinomial de iterações. Por isso, geralmente existe um nível mínimo ou máximo de melhorias a cada iteração que é adotado como critério de parada.

Tanto heurísticas construtivas quanto algoritmos de busca local são utilizados na construção de heurísticas modernas, mais conhecidas como meta-heurísticas, como veremos na próxima subseção.

## 2.3 Meta-Heurísticas

Estas são abordagens muito importantes para a resolução de problemas de otimização combinatória, como o problema do roteamento de veículos. Tratam-se de arcabouços gerais de regras formados a partir de um tema comum que podem servir de base para a construção de soluções aos mais diversos problemas computacionais [3]. As regras são escolhidas de acordo com o tema, arbitrariamente, ou seguindo limites estabelecidos pela própria natureza do problema, por exemplo, leis físicas, químicas, regras sociais. Assim, uma mesma meta-heurística é capaz de inspirar a criação de algoritmos para resolver problemas de áreas diversas.

Existem dois grandes grupos de meta-heurísticas, as de busca e as evolutivas. Exemplos das de busca são Busca Tabu e Busca em Vizinhança Variável. Estas meta-heurística generalizam a busca local, percorrendo as vizinhanças com diferen-

tes critérios. Já as meta-heurísticas evolutivas usam populações de indivíduos para representar variadas soluções e se inspiram na evolução natural para gerar novos indivíduos. Exemplo dessas são os algoritmos genéticos, meméticos e o BRKGA, pelo qual temos particular interesse.

## 2.4 Meta-heurística BRKGA

O Biased Random-Key Genetic Algorithm (BRKGA) é uma meta-heurística baseada em algoritmos genéticos, onde um conjunto de indivíduos avança utilizando o princípio evolutivo de Darwin ou seja, o princípio de sobrevivência do mais apto.

Cada indivíduo  $j$  de uma população é representado por um cromossomo  $Q^j$ , que corresponde a uma solução. Cada cromossomo consiste em um vetor de  $m$  alelos. Cada alelo é uma chave aleatória uniformemente escolhida no intervalo  $[0, 1]$ . Chamamos de decodificador um algoritmo que traduz um cromossomo  $Q^j$  em uma solução  $R^j$ . A adaptabilidade é uma função que avaliará a solução  $R^j$ .

O BRKGA inicializa com uma população de  $p$  cromossomos gerados aleatoriamente, e essa população evolui ao longo de  $g$  gerações. Para  $i = 1, \dots, g - 1$ , a população é atualizada da geração  $i$  para  $i + 1$  de acordo com os seguintes passos:

1. Para cada indivíduo  $j$  na geração  $i$ , o cromossomo  $Q^j$  é decodificado em  $R^j$ .
2. Avalia-se a adaptabilidade de cada solução  $R^j$  para cada indivíduo  $j$ .
3. São copiados os melhores  $p_e$  indivíduos (conjunto de elite) de  $i$  para  $i + 1$ .
4. São adicionados  $p_m$  cromossomos gerados aleatoriamente (mutantes) em  $i + 1$ .
5. Produz-se  $p - (p_e + p_m)$  novos cromossomos para  $i + 1$  usando cruzamentos.

Os cruzamentos geram novos indivíduos ao misturar alelos de dois cromossomos. Ambos os pais são da geração atual  $i$ , sendo exatamente um do conjunto de elite. A probabilidade de um alelo ser escolhido do pai elite é  $\rho_e$ . Após  $g$  gerações, o BRKGA retorna o cromossomo do melhor indivíduo  $Q^*$ , e sua solução decodificada  $R^*$ .

## 3 Objetivos

O objetivo principal deste projeto é o estudo do TSP-D, focando em técnicas heurísticas e meta-heurísticas, particularmente o BRKGA. Algoritmos para o TSP-D serão projetados utilizando as técnicas estudadas e os mais promissores serão implementados e validados experimentalmente, utilizando instâncias da literatura científica. Pretendemos apresentar os algoritmos projetados e as análises empíricas realizadas em uma conferência da área.

Além disso, esta iniciação científica irá introduzir o candidato a métodos de pesquisa e complementar a sua formação em Ciência da Computação, aprofundando seu conhecimento em problemas de otimização combinatória, heurísticas, meta-heurísticas e técnicas de projeto e análise de algoritmos.

## 4 Plano de Trabalho e Cronograma

Os primeiros quatro meses do projeto serão dedicados ao estudo do TSP-D e heurísticas gulosas e de busca local aplicáveis a ele, bem como ao desenvolvimento e implementação de algoritmos usando essas técnicas. Nos próximos quatro meses meta-heurísticas serão exploradas, com foco no BRKGA. Novamente, algoritmos serão projetados e o desempenho destes será avaliado empiricamente, através da implementação em linguagens como C/C++ ou Python. Destacamos que as heurísticas produzidas anteriormente poderão ser usadas nesta etapa. Nos últimos quatro meses do projeto focaremos na análise empírica dos resultados e em realizar ajustes e melhorias dos algoritmos a partir das informações obtidas nas análises. Havendo tempo, também gostaríamos de estudar e adaptar alguns dos nossos algoritmos para variantes do TSP-D.

Atividades	Meses											
	1	2	3	4	5	6	7	8	9	10	11	12
TSP-D e Heurísticas	•	•	•	•								
Meta-Heurística BRKGA					•	•	•	•				
Análises e Melhorias									•	•	•	•
Relatório técnico	•	•	•	•	•	•	•	•	•	•	•	•

## 5 Materiais e Métodos

Durante o projeto, o candidato estudará resultados importantes relacionados ao TSP-D, acessando esses artigos disponibilizados gratuitamente pela UFSCar, além dos livros sobre heurísticas e meta-heurísticas, disponibilizados pela biblioteca pública e pelo orientador.

No decorrer do projeto serão realizadas reuniões quinzenais entre o candidato e o orientador para averiguar o andamento do mesmo, discutir os conteúdos estudados, além de estabelecer direções e metas.

## Referências

- [1] Niels Agatz, Paul Bouman, and Marie Schmidt. Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, 52(4):965–981, 2018.
- [2] Sanjoy Dasgupta, Christos H Papadimitriou, and Umesh Virkumar Vazirani. *Algorithms*. McGraw-Hill Higher Education New York, 2008.
- [3] Fred W. Glover and Gary A. Kochenberger. *Handbook of metaheuristics*, volume 57. Springer Science & Business Media, 2006.
- [4] José Fernando Gonçalves and Mauricio GC Resende. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525, 2011.
- [5] Quang Minh Ha, Yves Deville, Quang Dung Pham, and Minh Hoàng Hà. On the min-cost traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies*, 86:597–621, 2018.
- [6] Ines Khoufi, Anis Laouiti, and Cedric Adjih. A survey of recent extended variants of the traveling salesman and vehicle routing problems for unmanned aerial vehicles. *Drones*, 3(3):66, 2019.
- [7] Patchara Kitjacharoenchai, Mario Ventresca, Mohammad Moshref-Javadi, Seokcheon Lee, Jose MA Tanchoco, and Patrick A Brunese. Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. *Computers & Industrial Engineering*, 129:14–30, 2019.
- [8] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, Young-Han Nam, and Mérouane Debbah. A tutorial on uavs for wireless networks: Applications, challenges, and open problems. *IEEE communications surveys & tutorials*, 21(3):2334–2360, 2019.
- [9] El-Ghazali Talbi. *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons, 2009.
- [10] Phan Anh Tu, Nguyen Tuan Dat, and Pham Quang Dung. Traveling salesman problem with multiple drones. In *Proceedings of the Ninth International Symposium on Information and Communication Technology*, pages 46–53, 2018.