

Universidade Federal de São Carlos

Processo FAPESP #2020/06103-0

Heurísticas e Meta-Heurísticas para Problemas de Roteamento de Veículos com Drones

Relatório Parcial

Orientador: Prof. Dr. Mário César San Felice

Bolsista: Matheus Teixeira Mattioli

Agosto de 2021 a Janeiro de 2022
São Carlos - SP, Brasil

Sumário

1	Introdução	2
2	Hierarquia de Problemas com Drones	2
3	Definição do Problema Escolhido	4
4	Algoritmos para o TSP-D	5
4.1	Algoritmos Propostos	8
5	Cronograma	9
6	Conclusão	9

1 Introdução

Neste relatório parcial vamos tratar de problemas de otimização combinatória, área da computação responsável por estudar problemas em que é necessário tomar decisões que maximizem ou minimizem uma função objetivo. Em particular, escolhemos problemas de roteamento de veículos com drones, nos quais devemos traçar rotas conectando pontos de interesse por meio de veículos tripulados (caminhões) e veículos não tripulados (drones). Apesar dessa definição simples, na prática este problema pode ser aplicado para uma gama muito grande de serviços, por exemplo, podemos realizar monitoramento de fronteiras ou áreas florestais, realizar serviços urbanos, como leitura de consumo de água e energia, ou entrega de produtos à clientes. Este último, em particular interessa a grandes empresas como a Amazon, segundo Shiva et al. [18].

Este problema também desperta interesses acadêmicos, pois além das dificuldades inerentes ao VRP e TSP, existe a complexidade de se trabalhar com drones, conforme apontam Sugihara e Gupta [19]. Algumas dessas dificuldades são o tempo de duração da bateria dos drones e a carga que o drone aguenta transportar. Assim, ao combinar os dois tipos de transporte, temos um problema bastante desafiador. Para resolver essa situação, muitos pesquisadores apostaram no uso de heurísticas e meta-heurísticas, obtivendo resultados interessantes, conforme vemos nos artigos de Quang Minh et al. [8], Murray [15], Sundar [20] e Freitas et al. [7]. Por conta disso, resolvemos utilizar heurísticas e metaheurísticas para atacar problemas de roteamento de drones nesta Iniciação Científica.

Problemas de roteamento com drones podem ser encontrados com muitos nomes distintos na literatura e possuem diversas variantes. Por conta disso, resolvemos realizar um levantamento bibliográfico. O survey de Khoufi et al. [10] nos conta sobre problemas de *Path Planning* envolvendo drones, tais como monitoramento de grandes áreas por drones utilizando bases de recarregamento de bateria. Já o survey de Mozaffari et al. [14] nos conta, principalmente, sobre os benefícios do uso de drones para comunicação wireless, visando manutenção de redes ou coleta de informações. Neste contexto o drone é comumente chamado de *data mule*. Nas próximas seções descreveremos com mais detalhes a classificação encontrada para estes problemas de roteamento envolvendo drones.

Nesse relatório serão apresentadas as atividades realizadas na primeira metade do projeto. Na Seção 2 será descrito a organização de problemas com drones encontrada. A Seção 3 descreve o problema que pretendemos atacar. A Seção 4 descreve heurísticas e metaheurísticas que estudamos e pretendemos utilizar. A Seção 5 apresenta o cronograma da iniciação científica. Por fim, a Seção 6 conclui o relatório técnico, explicitando os principais aprendizados deste primeiro semestre de iniciação científica.

2 Hierarquia de Problemas com Drones

Nesta seção vamos classificar os problemas de roteamento envolvendo drones conforme estudados nos surveys citados anteriormente. Dessa forma, temos a seguinte hierarquia de problemas com drones:

Otimização de trajetória (*Trajectory Optimization*). Consiste em identificar uma trajetória que minimize alguma medida de desempenho, como tempo de voo ou consumo de combustível, enquanto satisfaz um conjunto de restrições cinemáticas, por

exemplo, posição, velocidade e aceleração, e restrições dinâmicas, como forças e momento do veículo [5].

Planejamento de rota (*Path Planning*). Reduz o nível de detalhes físicos, por exemplo, ignora a dinâmica do veículo, mas ainda leva em consideração a posição e geometria dos drones e objetos do ambiente[16, 17]. Ocasionalmente, variantes do *Path Planning* podem conservar alguma dinâmica do veículo, por exemplo, restrições de movimento ou vento [13, 12].

Problemas de Roteamento de Drones (*UAV¹ Routing Problems*). Caixaieiro viajante (TSP), Roteamento de veículos (VRP) e variantes envolvendo drones.

Atribuição de tarefas à drones (*UAV Task Assignment*). É um subproblema de Problemas de Roteamento, em que se deseja encontrar a atribuição ideal de tarefas em vez de atribuir mercadorias [9]. Permite múltiplas visitas ou subtours.

TSP com drones (TSP-D). Categorias de TSP-D são descritas a seguir:

Entrega-Transporte (*Delivery-Transportation*). Realizamos o roteamento de veículos e drones. Mais especificamente, roteamos o veículo tripulado até certo ponto e, nesse ponto, lançamos o drone para realizar tarefas em outros nós que o veículo não visita. Também conhecido como *last-mile delivery*. Pode ser Delivery 1-Truck 1-Drone [1], 1-Truck m-Drone [21], n-Truck m-Drone [11], de acordo com o número de caminhões e drones envolvidos.

Delay Tolerant Networks (DTN). As DTN são redes esparsas onde a comunicação direta de ponta a ponta completa entre origem e destino raramente pode ser estabelecida [2]. Os mecanismos de roteamento em DTNs contam com a mobilidade dos nós para conectar nós desconectados, transportando mensagens pela rede para superar a desconexão do caminho. A abordagem DTN consiste em introduzir nós dedicados em estabelecer comunicação entre nós comuns e aliviá-los de trabalhos que consomem energia, como roteamento e encaminhamento de mensagens. Os Veículos Aéreos Não Tripulados (UAVs) têm muitas aplicações potenciais em sistemas de comunicação sem fio, como fornecer conectividade sem fio econômica para dispositivos sem cobertura de infraestrutura devido, por exemplo, ao sombreamento intenso por terreno urbano ou montanhoso, ou danos à infraestrutura de comunicação devido a desastres naturais. Neste contexto, os UAVs podem ser nomeados de Data Mule.

Inteligência, Vigilância e Reconhecimento. Queremos atender determinadas regiões do espaço com o drone, porém devemos recarregá-lo em determinados pontos de recarga. Um exemplo é o problema The Fuel Constrained UAV Routing Problem (FCURP) [20].

Monitoramento, Rastreamento, Filmagem. Dentre muitos exemplos, esta aplicação pode ter motivações militares de monitoramento de fronteiras

¹Unmanned Aerial Vehicle.

ou científica, para filmagem em áreas de floresta. Um problema que trata desse assunto é o Close Enough Travelling Salesman Problem (CETSP) [6]. No CETSP, o caixeiro (no caso o drone) deve visitar uma região específica que contenha o vértice e não o próprio vértice. Em várias aplicações dessa categoria os UAVs não precisam chegar ao local de destino. Assim, o caminho dele, bem como a energia de sua bateria, podem ser otimizados. Quando o drone está equipado com um sensor, ele pode operar com sucesso a uma certa distância do alvo.

VRP com drones (VRP-D). Classificações para VRP-D são apresentadas abaixo:

Entrega-Transporte (*Delivery-Transportation*). Semelhante ao descrito anteriormente, porém agora os drones e os caminhões podem fazer parte de um VRP. Exemplo, Delivery por n-Trucks e m-Drones, uma frota de caminhões equipados com drones entregam bens aos clientes [22].

Entrega por Drones. Entrega feita apenas por drones, o que é desafiador, devido a múltiplas características operacionais como operações “multi-trip”, planejamento de recarga e cálculo de consumo de energia. Exemplo: Multi-Trip Drone Routing Problem (MTDRP) [4], considera a influência da carga útil e da distância na duração do voo.

Monitoramento, Vigilância. Podemos usar uma frota de drones para combater incêndios florestais [3], que são uma classe de desastres perigosos que são muito difíceis de combater. Especialmente incêndios em terrenos montanhosos ao usar equipamentos tradicionais. Dada a rapidez dos incêndios florestais, a atribuição adaptativa e dinâmica de tarefas de combate a incêndios para drones é de grande importância.

3 Definição do Problema Escolhido

Dadas as definições encontradas na literatura, escolhemos atacar um problema TSP-D, mais especificamente o Delivery-Transportation 1-Truck 1-Drone, que chamaremos apenas de TSP-D neste relatório. Nele, optamos por considerar que o caminhão nunca está parado, então em um certo nó, lançamos o drone, que estava no caminhão, para realizar uma entrega e retornar em um ponto de encontro, onde o caminhão passa e coleta o drone. Dessa forma, temos dois conjuntos, um conjunto de pontos atendidos pelo caminhão TD e um conjunto de pontos atendidos pelo drone DD , a solução é formada pelos dois conjuntos $sol = \{TD, DD\}$, sendo que as entregas feitas pelos drones têm que respeitar a capacidade e bateria do drone e são indicadas na forma de tripla $\langle i, j, k \rangle$, sendo i o ponto de lançamento, j o ponto atendido pelo drone e k o ponto de encontro. Imagens para ilustrar um tour ótimo para o TSP e para o TSP-D são apresentadas nas Figuras 1 e 2 retiradas do artigo de Quang Minh et al. [8].

Mais formalmente, dado um grafo $G = (V, A)$, com $V = \{0, 1, \dots, n, n+1\}$, sendo 0 e $n+1$ o mesmo depósito, $N = \{1, \dots, n\}$, queremos encontrar uma rota que comece com o caminhão e o drone em 0 e termine com os dois em $n+1$ de forma a minimizar

$$C1 \sum_{i \in V_L} \sum_{j \in V_R: i \neq j} d_{ij} x_{ij} + C2 \sum_{i \in V_L} \sum_{j \in N, i \neq j} \sum_{k \in V_R: \langle i, j, k \rangle \in P} (d'_{ij} + d'_{jk}) y_{ijk} ,$$

sendo $C1$ os custos operacionais do caminhão, $C2$ os custos operacionais do drone, $V_L = \{0, 1, \dots, n\}$, $V_R = \{1, 2, \dots, n + 1\}$, $x_{ij} \in \{0, 1\}$, $y_{ijk} \in \{0, 1\}$, d_{ij} a distância percorrida pelo caminhão entre os pares de vértices i, j e d'_{ij} a distância percorrida pelo drone entre i, j . Uma solução deve respeitar as seguintes restrições:

- Cada cliente é atendido apenas uma vez pelo caminhão ou pelo drone.
- As entregas feitas pelo drone têm que respeitar sua capacidade e bateria.
- As entregas feitas pelo drone devem ser compatíveis com a rota do caminhão, isto é, para cada ponto de lançamento deve existir um ponto de encontro e o drone só pode ser lançado novamente após sua coleta nesse ponto de encontro.

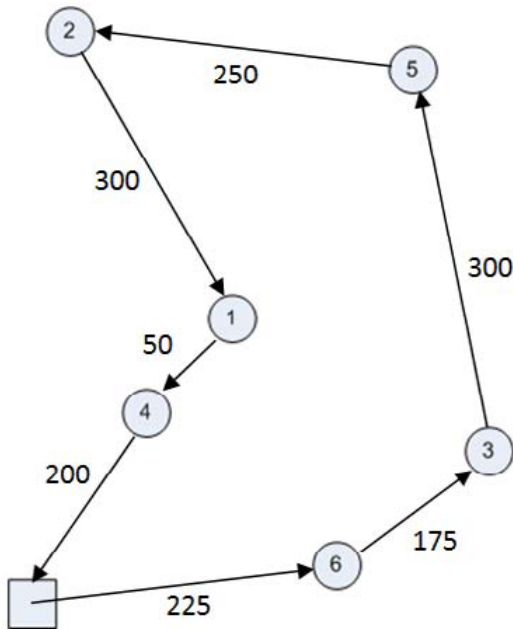


Figura 1: Solução ótima para um TSP. Imagem retirada do artigo [8].

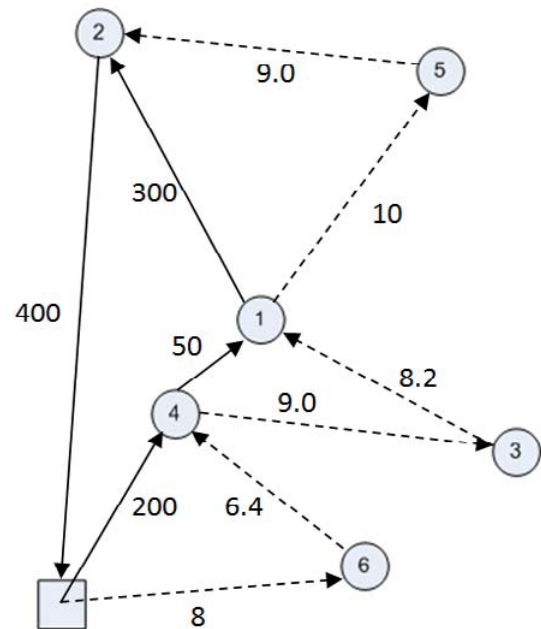


Figura 2: Solução ótima para um TSP-D. Imagem retirada do artigo [8].

4 Algoritmos para o TSP-D

Na iniciação científica anterior estudamos e implementamos algoritmos heurísticos e metaheurísticos para o TSP e o VRP, alguns dos quais desejamos adaptar para o TSP-D. Assim, estudamos algumas heurísticas específicas para o TSP-D e buscamos formas de adaptar as ideias estudadas anteriormente para este novo problema. Ao longo desta seção descreveremos melhor os procedimentos que estudamos no trabalho de Quang Minh et al. [8].

O principal algoritmo estudado é um *Greedy Randomized Adaptive Search Procedure* (GRASP). A cada iteração do algoritmo, realizamos um procedimento construtivo aleatorizado seguido de uma intensificação, a busca local. O procedimento construtivo é um

Algorithm 1 GRASP

Input: Grafo $G = (V, A)$, funções de distância d e d' e fatores de custo $C1$ e $C2$

Output: Uma solução sol^*

```
1: function GRASP( $G$ )
2:    $sol^* \leftarrow +\infty$ 
3:   while Critério de parada não atingido do
4:      $tspTour \leftarrow greedyRandTSP(G)$ 
5:      $P, C, D \leftarrow makeAuxGraph(tspTour)$ 
6:      $S \leftarrow solveTspD(P, C, D)$ 
7:      $sol \leftarrow localSearch(S)$ 
8:     if  $sol$  for melhor que  $sol^*$  then
9:        $sol^* \leftarrow S$ 
10:  return  $sol^*$ 
```

algoritmo guloso aleatorizado para o TSP seguido de um algoritmo para transformar essa solução do TSP em uma do TSP-D.

No Algoritmo 1 temos o pseudocódigo do algoritmo estudado. Para satisfazer a função *greedyRandTSP* podemos implementar diversos algoritmos aleatorizados. Alguns exemplos citados no artigo [8] são:

Nearest Neighbor Aleatorizado. Nele escolhemos o vértice inicial aleatoriamente e em cada iteração, escolhemos o próximo vértice a ser inserido de forma pseudoa-leatória da Lista Restrita de Candidatos (LRC). Na LRC colocamos os k vizinhos mais próximos do último vértice inserido, com k sendo um parâmetro que indica do algoritmo.

Permutação Aleatória. O algoritmo escolhe aleatoriamente a ordem de visita dos vértices.

Cheapest Insertion Aleatorizado. A abordagem consiste em começar com um sub-tour, ou seja, um tour com um subconjunto de vértices do grafo, e então estender este tour inserindo repetidamente os nós restantes até que nenhum outro possa ser adicionado. O vértice não visitado v a ser inserido e sua localização entre dois nós consecutivos (i, j) do passeio são selecionados de modo que esta combinação forneça o menor Custo de Inserção (CI). Este custo é calculado por:

$$CI = d_{iv} + d_{vj} - d_{ij} .$$

Para adicionar aleatoriedade nessa heurística, a cada iteração, escolhemos um vértice não visitado e sua localização de inserção usando uma LRC.

A função *makeAuxGraph* constrói um grafo auxiliar orientado a partir do circuito do TSP e retorna os vetores P , C e D . O grafo auxiliar é constituído pelo circuito do TSP e algumas arestas adicionais. O procedimento considera cada par de vértices (i, k) , com i estando antes de k no circuito. Fixados (i, k) o algoritmo escolhe um j entre i e k para ser atendido pelo drone de modo a minimizar o custo total de entrega. Então, a tripla $\langle i, j, k \rangle$ é inserida no vetor D , que indica as possíveis entregas por drone, e uma aresta adicional a_{ik} é inserida no grafo auxiliar. Essa aresta simboliza que o drone é lançado

em i e o caminhão realiza entregas em todos os vértices até k , com exceção do vértice intermediário j . A Figura 3 ilustra o grafo auxiliar formado por esse procedimento a partir do circuito da Figura 1.

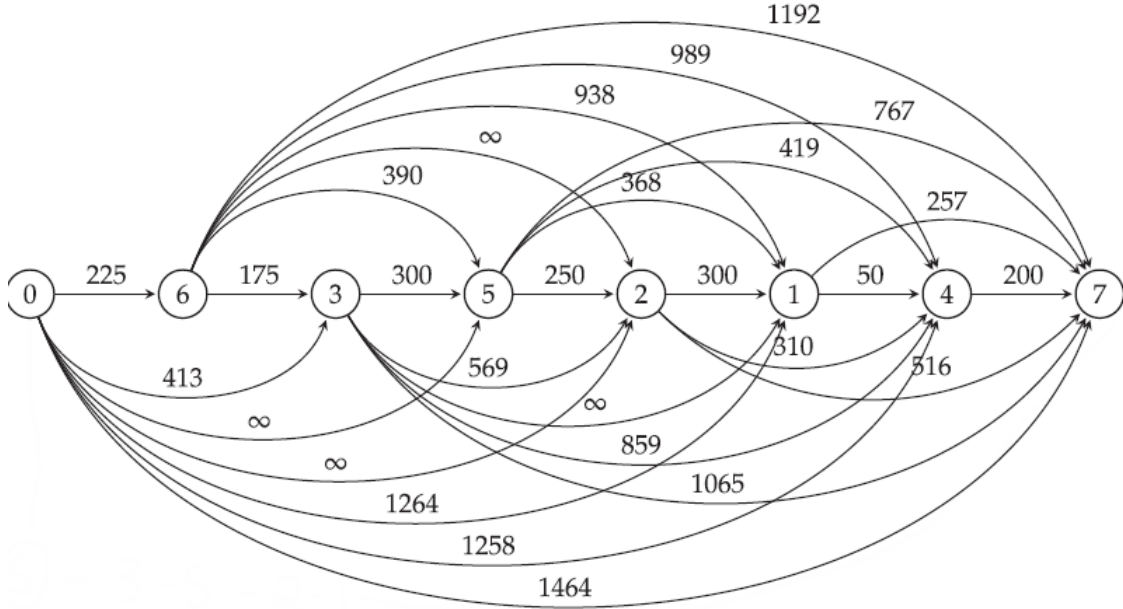


Figura 3: As arestas na horizontal correspondem ao circuito original, as outras são as adicionadas por makeAuxGraph. Uma a_{ik} com custo ∞ indica que não é possível uma entrega com o drone saindo de i e voltando em k . Figura retirada do artigo [8].

Por fim, a função makeAuxGraph encontra o caminho mais curto de cada vértice do grafo auxiliar e armazena no vetor P , com $P[n+1]$ representando o “caminho” mais curto que começa e termina no depósito. Os custos de cada caminho são armazenados no vetor C .

A partir dos vetores P , C e D , a função solveTspD transforma o caminho mais curto do grafo auxiliar na solução $sol = \{TD, DD\}$. Isso ocorre por meio da análise do caminho mais curto $P[n+1]$, que é separado em *Truck Deliveries* (TD) e *Drone Deliveries* (DD) de acordo com as arestas utilizadas. Em particular essa etapa usa vetor de triplas D para identificar as DDs e separar das TDs no caminho mais curto.

Na fase de intensificação é utilizada uma busca local. As estruturas de vizinhança apresentadas pelo artigo são *Exchange*, também conhecida como *Swap*, e *Relocation*, adaptadas ao TSP-D. No TSP, a estrutura de vizinhança *Exchange* funciona através da seleção de dois vértices da solução inicial e troca de suas posições no circuito. Por exemplo, se realizarmos um *exchange* entre 0 e 3 no tour inicial $\{0, 1, 2, 3, 4, 5\}$, obtemos o tour $\{3, 1, 2, 0, 4, 5\}$. E a estrutura *Relocation* no TSP, funciona realocando um vértice x para ser predecessor de um vértice y . Como exemplo, considerando o tour anterior e $x = 1$ e $y = 4$, obtemos $\{0, 2, 3, 1, 4, 5\}$.

O *Exchange* para TSP-D realiza as seguintes operações de troca: (1) Um nó em DD com um em TD; (2) Dois nós em DD; (3) Dois nós em TD. Em relação a operação (1) atualizamos a tripla $\langle i, j, k \rangle$ de DD substituindo o vértice j por x e removemos x de TD, inserindo j na sua posição. A operação (2) é simples, apenas realizamos uma troca entre

um j e outro j' nas triplas do vetor DD. Já a operação (3) troca dois vértices do vetor TD, tomando os devidos cuidados para atualizar as triplas de DD se algum dos vértices trocados fizer parte de alguma tripla.

O *Relocation* é dividido em duas operações, *Truck-Relocation* e *Drone-Relocation*. A primeira operação funciona de forma semelhante ao *Relocation* do TSP, pois consideramos que as realocações só podem ocorrer entre vértices exclusivos ao TD, isto é, um vértice atendido pelo caminhão que não participe de nenhuma tripla de DD. Já no *Drone-Relocation* a ideia é transformar um vértice de TD em um de DD, formando uma tripla nova, ou alterar os pontos de lançamento i e coleta k de um vértice j . As duas operações podem ser realizadas ao mesmo tempo também. Essa estrutura pode resultar em uma vizinhança que apresente mais *drone deliveries* que a solução inicial.

Outra estrutura própria para TSP-D apresentada pelo artigo é o *Drone Removal*, no qual removemos uma tripla de DD e o vértice j passa a ser atendido pelo caminhão.

4.1 Algoritmos Propostos

A partir desse estudo, propusemos alguns algoritmos que podem trazer melhores resultados para o TSP-D. Em particular, pensamos em algumas heurísticas para a fase construtiva, apresentadas a seguir:

Inserção de Bicos. Neste algoritmo queremos inserir o máximo de triângulos acutângulos no circuito do TSP. Uma forma de realizar esta operação é, a partir de uma escolha inicial aleatória, escolher como próximo vértice a ser visitado o segundo mais próximo do último inserido. A ideia é que esses “bicos” levam a boas oportunidades de entrega por drone.

Retirada de Bicos. Neste algoritmo queremos remover triângulos acutângulos do circuito do TSP da seguinte forma: selecionamos o vértice mais distante do centróide do grafo e consideramos que este vértice é atendido por um drone, então, recalculamos o centróide. A ideia é formar um TSP nos vértices que sobraram e utilizar os vértices marcados como entrega de drone já na etapa de resolução do TSP-D. As retiradas podem parar de acontecer se nenhum vértice estiver acima de uma margem de distanciamento estipulada.

Expansão do Grafo Auxiliar. Queremos expandir a construção do grafo auxiliar com o objetivo de não limitar o vértice j a estar entre i e k no circuito do TSP. Para isso, percorremos os vértices do circuito e, a cada iteração, consideramos que o nó corrente é atendido pelo drone. Escolhemos um ponto i de lançamento e um k de coleta de forma gulosa, respeitando apenas a restrição da posição de k ser maior que a de i . Dessa forma, criamos caminhos adicionais que armazenam essa tripla. O restante do grafo auxiliar é construído como antes. No final, obtemos n caminhos adicionais, e então construímos n vetores P, C, D para serem considerados na formulação da solução.

Na fase de intensificação pensamos em utilizar a metaheurística VND (*Variable Neighborhood Descent*) para substituir a heurística de busca local. A VND usa várias vizinhanças, em ordem crescente de complexidade. Quando uma estrutura de vizinhança

menos complexa alcança o mínimo local, avançamos para uma mais complexa. Se esta conseguir algum resultado que melhore a solução, retornamos à estrutura anterior e continuamos a busca com ela. O algoritmo termina quando alcançamos o mínimo local em todas as estruturas. No caso, as estruturas que pensamos em utilizar são as mesmas apresentadas pelo artigo estudado além de uma adaptação da vizinhança 2-OPT para o TSP-D, que queremos propor.

5 Cronograma

A Tabela 1 contém o cronograma proposto.

Tabela 1: Cronograma para as próximas etapas.

Atividades	Meses											
	1	2	3	4	5	6	7	8	9	10	11	12
Classificação dos Problemas	•	•	•									
Heurísticas				•	•	•	•					
Meta-Heurísticas								•	•	•	•	•
Implementação	•	•	•	•	•	•	•	•	•	•	•	•

O primeiro semestre foi destinado ao estudo e compreensão dos problemas de roteamento envolvendo drones. Destacamos que esse processo levou mais tempo que o esperado, por conta da variedade de problemas e da literatura ser recente, o que acarreta grande quantidade de problemas semelhantes e inconsistências na nomenclatura. Apesar disso, o processo de revisão bibliográfica trouxe um aprendizado importante para o estudante. A partir da compreensão dos problemas, buscamos escolher um problema viável para ser atacado durante o tempo da IC. Decidimos pelo TSP-D, por ele ser próximo ao TSP, ao mesmo tempo que apresenta relevância prática e um desafio interessante. Por conta dos conhecimentos sobre heurísticas e metaheurísticas adquiridas na iniciação anterior, escolhemos aprofundar no artigo de Quang Minh et al. [8] que aborda TSP-D com GRASP.

Feita esta escolha, pensamos em estender o trabalho estudado usando um GRASP-VND com novas heurísticas gulosas aleatorizadas. Nas próximas etapas do projeto pretendemos implementar e testar algumas dessas nossas ideias.

6 Conclusão

Nos primeiros meses desta iniciação científica buscamos entender a área de roteamento de veículos com drones e enfrentamos algumas dificuldades para entender a nomenclatura desta área. Por isso, partimos para o estudo de surveys que nos ajudaram a entender a variedade e classificação dos problemas, principalmente, o survey de Khoufi et al. [10]. Dos problemas encontrados nessa pesquisa, escolhemos o problema TSP-D para atacar. Em seguida estudamos heurísticas e metaheurísticas específicas deste problema e projetamos novas, as quais pretendemos implementar o mais breve possível.

As reuniões com o orientador foram feitas quinzenalmente, com exceção do mês de janeiro, em que foram semanais. Ainda não obtivemos resultados empíricos, porém considero satisfatório a ampla revisão da literatura realizada para entender as classificações de problemas de roteamento com drones, pois adquiri bastante conhecimento em problemas NP-Difíceis e pesquisa científica de artigos e surveys.

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] Cintia Barroca, António Grilo, and Paulo Rogério Pereira. Improving message delivery in uav-based delay tolerant networks. In *2018 16th International Conference on Intelligent Transportation Systems Telecommunications (ITST)*, pages 1–7. IEEE, 2018.
- [3] Kun Chen, Qibo Sun, Ao Zhou, and Shangguang Wang. Adaptive multiple task assignments for uavs using discrete particle swarm optimization. In *International Conference on Internet of Vehicles*, pages 220–229. Springer, 2018.
- [4] Chun Cheng, Yossiri Adulyasak, and Louis-Martin Rousseau. *Formulations and exact algorithms for drone routing problem*. CIRRELT, Centre interuniversitaire de recherche sur les réseaux d’entreprise ..., 2018.
- [5] Walton Pereira Coutinho, Maria Battarra, and Jörg Fliege. The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review. *Computers & Industrial Engineering*, 120:116–128, 2018.
- [6] Walton Pereira Coutinho, Roberto Quirino do Nascimento, Artur Alves Pessoa, and Anand Subramanian. A branch-and-bound algorithm for the close-enough traveling salesman problem. *INFORMS Journal on Computing*, 28(4):752–765, 2016.
- [7] Júlia Cária de Freitas and Puca Huachi Vaz Penna. A variable neighborhood search for flying sidekick traveling salesman problem. *International Transactions in Operational Research*, 27(1):267–290, 2020.
- [8] 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.
- [9] Xiaowei Jiang, Qiang Zhou, and Ying Ye. Method of task assignment for uav based on particle swarm optimization in logistics. In *Proceedings of the 2017 international conference on intelligent systems, metaheuristics & swarm intelligence*, pages 113–117, 2017.
- [10] 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.

- [11] 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.
- [12] He Luo, Zhengzheng Liang, Moning Zhu, Xiaoxuan Hu, and Guoqiang Wang. Integrated optimization of unmanned aerial vehicle task allocation and path planning under steady wind. *Plos one*, 13(3):e0194690, 2018.
- [13] Douglas G Macharet and Mario FM Campos. A survey on routing problems and robotic systems. *Robotica*, 36(12):1781–1803, 2018.
- [14] 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.
- [15] Chase C Murray and Amanda G Chu. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54:86–109, 2015.
- [16] Mohammadreza Radmanesh, Manish Kumar, Paul H Guentert, and Mohammad Sarim. Overview of path-planning and obstacle avoidance algorithms for uavs: A comparative study. *Unmanned systems*, 6(02):95–118, 2018.
- [17] S Sarath Chandra and ASCS Sastry. Prototype survey of path planning and obstacle avoidance in uav systems. *Int. J. Eng. Technol.(UAE)*, 7:316–322, 2018.
- [18] Shiva Ram Reddy Singireddy and Tugrul U Daim. Technology roadmap: Drone delivery–amazon prime air. In *Infrastructure and Technology Management*, pages 387–412. Springer, 2018.
- [19] Ryo Sugihara and Rajesh K Gupta. Path planning of data mules in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 8(1):1–27, 2011.
- [20] Kaarthik Sundar and Sivakumar Rathinam. Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots. *IEEE Transactions on Automation Science and Engineering*, 11(1):287–294, 2013.
- [21] 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.
- [22] Xingyin Wang, Stefan Poikonen, and Bruce Golden. The vehicle routing problem with drones: several worst-case results. *Optimization Letters*, 11(4):679–697, 2017.