Um algoritmo eficiente para um problema multiobjetivo de roteamento em rede de VANTs

Elias L. Marques Jr.[1] **, Vitor N. Coelho**[2] **, Igor M. Coelho**[1] **, Bruno N. Coelho** [3] **, Luiz Satoru Ochi** [1]

[1]Instituto de Computação – Universidade Federal Fluminense (UFF)
Niterói – RJ – Brasil

[2]OptBlocks Consultoria Ltda.
Avenida João Pinheiro, 274 Sala 201 – 30130-186 Belo Horizonte – MG – Brasil

[3]Departamento de Engenharia de Controle e Automação – UFOP
35400-000 – Ouro Preto – MG – Brazil

eliaslawrence@id.uff.br,vncoelho@gmail.com,brunonazario@ufop.edu.br,{imcoelho,satoru}@ic.uff.br

Abstract. *This paper deals with Unmanned Aerial Vehicle (UAV) routing in dynamic grid scenarios with limited battery autonomy and multiple charging stations. The problem is inspired by real-world constraints, specially designed for overcoming challenges of a limited vehicle driving range. A multi-objective variant of Variable Neighborhood Search (VNS) is considered for finding a set of non-dominated solutions, while respecting the navigation over forbidden areas and also battery capacity. A case of study was developed where one UAV has to attend clients spread throughout a grid representing a map.*

Resumo. *Este artigo trata do roteamento de Veículos Aéreos Não Tripulados (VANT) em cenários dinâmicos de rede com autonomia de bateria limitada e múltiplas estações de carregamento. O problema é inspirado em restrições do mundo real, especialmente projetado para superar os desafios de um alcance limitado de condução de veículos. Considera-se uma variante multiobjetivo do Variable Neighborhood Search (VNS) para encontrar um conjunto de soluções não dominadas, respeitando a navegação em áreas proibidas e também a capacidade da bateria. Foi desenvolvido um caso de estudo onde um VANT deve atender clientes espalhados por uma rede representando um mapa.*

1. Introdução

Diferentes áreas de conhecimento incluindo o segmento de cidades e regiões inteligentes tem se beneficado de inovações tecnológicas como a miniaturização de sistemas de controle eletrônico e a redução de custos de componentes eletrônicos [Floreano and Wood 2015] que resultaram em um aumento na disponibilidade de Veículos Aéreos Não Tripulados (VANTs), também conhecidos como Drones, ou Sistemas Aéreos Não Tripulados (UAS) .

Embora o drone seja frequentemente relacionado a entusiastas, entretenimento e indústria fotográfica, seu uso tem se multiplicado em aplicações militares, civis e comerciais. Vigilância aérea, reconhecimento e rastreamento de objetos são algumas das muitas

outras aplicações que estão surgindo com potencial para uso de VANTs. Inúmeros outros podem surgir da criatividade humana em um futuro próximo [Coelho et al. 2017]. Já existem alguns trabalhos que mostram aplicações na vida cotidiana incluindo por exemplo:

- Inspeção de infraestrutura [Metni and Hamel 2007], [Máthé and Buşoniu 2015], [Irizarry et al. 2012];
- Inspeção de Linha Elétrica [Adabo 2014], [Deng et al. 2014];
- A vigilância de um espaço alvo usando veículos aéreos é um tópico de interesse de pesquisa atual para aplicações como monitoramento do clima, levantamentos geográficos e talvez exploração extraterrestre [Nigam and Kroo 2008];
- A grande flexibilidade dos VANTs pode possibilitar novas abordagens durante a coleta de dados de sensoriamento remoto como, por exemplo, integrar o mapeamento em tempo real e navegação autônoma [Haala et al. 2011];
- O monitoramento ambiental [Harris et al. 2005].

O setor de transporte de cargas, em particular, já mostra interesse e investimentos em aplicações de VANTs. O crescimento do e-commerce tem sustentado esse interesse de grandes empresas. Os drones são capazes de decolar e pousar com segurança nas proximidades de prédios e humanos, melhorando a qualidade do serviço atual em áreas congestionadas ou remotas [Floreano and Wood 2015].

Quando se discute o serviço de entrega de mercadorias, estamos, implicitamente falando de um Problema de Roteamento de Veículos - VRP e suas variantes, por exemplo. O que significa brevemente um problema de projetar rotas ótimas de um ou vários depósitos para vários clientes ou pontos estratégicos geograficamente dispersos, sujeitos a restrições espaciais e temporais [Laporte 1992].

Embora existam muitos trabalhos na literatura relacionados às variações do VRP [Gutin and Punnen 2006], os que abordam o roteamento de VANTs ainda são poucos como o TSPD (Traveling Salesman Problem with Drone) [Agatz et al. 2018], o problema de roteamento de veículos com drones [Wang et al. 2017] e a abordagem VNS de Schermer et al. [Schermer et al. 2018].

No entanto, não podemos focar apenas nos avanços tecnológicos e simplesmente esquecer os danos ao meio ambiente que eles podem causar. É por isso que a comunidade científica tem se preocupado tanto com o desenvolvimento de tecnologias verdes e isso não difere na computação. O Green Vehicle Routing Problem (G-VRP), por exemplo, proposto por Erdoğan et al.[Erdoğan and Miller-Hooks 2012], adiciona ao VRP original, restrições sobre economia de combustível.

O TSP com seleção de hotéis [Vansteenwegen et al. 2012, de Sousa et al. 2021] é uma variação do TSP com semelhanças com o problema abordado neste artigo. O objetivo principal é minimizar o número de viagens e tempo total percorrido. Esse problema é encontrado em cenários reais como entrega de produtos por veículos elétricos que precisam ser recarregados ao longo de uma rota.

Para abordar aplicações reais complexas e muitas vezes dinâmicas, em muitos casos é necessário abordá-las como um problema multiobjetivo que forneça um conjunto de soluções não dominadas com diferentes rotas e itinerários possíveis. Quanto mais

funções objetivas e restrições, usualmente ficamos mais próximo aos modelos do mundo real, mas também mais complexo o problema se torna.

As principais contribuições deste trabalho atual são:

- Proposta de um novo algoritmo heurístico para o problema de roteamento de VANTs dependente do tempo [Coelho et al. 2017], e considerando as seguintes condições:
  - respeitando os requisitos operacionais dos VANTs;
  - abordando o microespaço aéreo considerando um cenário de inspeção de pontos, e evitando pontos proibidos (restrições de atracação) [Coelho et al. 2016];
  - integrando os VANTs aos novos conceitos de sistemas de mini/microrredes, nos quais os veículos podem ser carregados em diferentes pontos das futuras cidades inteligentes;
  - rotas dinâmicas considerando drones já em movimento: instâncias com bateria inicial diferente de 100%, ponto de origem aleatório e número de clientes já visitados.

O restante deste artigo está organizado da seguinte forma. A Seção 2 descreve o modelo proposto e o intervalo de parâmetros reais considerados em nossas análises, enquanto a Seção 3 contém a metodologia empregada para resolver o problema. Na Seção 4, encontram-se os experimentos computacionais comparando as diferentes implementações, instâncias, variáveis e resultados. Por fim, a Seção 5 conclui o trabalho e apresenta direções de pesquisas futuras.

2. Descrição do Problema

O caso de estudo aqui projetado é composto por um espaço aéreo dividido em faixas horizontais e verticais, onde o veículo pode se deslocar seguindo a distância de Chebyshev, onde as distâncias entre quaisquer pontos adjacentes são as mesmas. As estações de energia estão espalhadas na área de roteamento e acessadas pelo drone para recarregar suas baterias. Para representar áreas proibidas, a grade também é composta por pontos proibidos que o VANT não pode acessar, caso contrário invalidaria a rota.

Como problema de roteamento, o veículo deve atender clientes que estão espalhados pela rede. Para isso, o ponto correspondente ao cliente deve fazer parte da rota final. Isso significa que as coordenadas x e y do cliente devem fazer parte do array que representa a solução. Na Seção 3 discutimos uma alternativa para pré-processar distâncias mais curtas e armazená-las em estrutura de dados auxiliar, de modo que apenas os pontos de origem-destino e as distâncias precisam ser considerados entre os pontos de carga e entrega. No entanto, devido à natureza dinâmica do problema, pode ser necessário processá-los novamente, após ocorrerem alterações nos dados de entrada.

O trabalho atual considera que os dados dinâmicos são passados como entrada, de modo que nenhuma alteração precisa ser realizada durante a pesquisa. Como as instâncias já consideram a localização e a capacidade inicial arbitrária do drone (carga da bateria), uma variante dependente do tempo pode ser considerada como uma extensão deste trabalho (consulte a Seção 5).

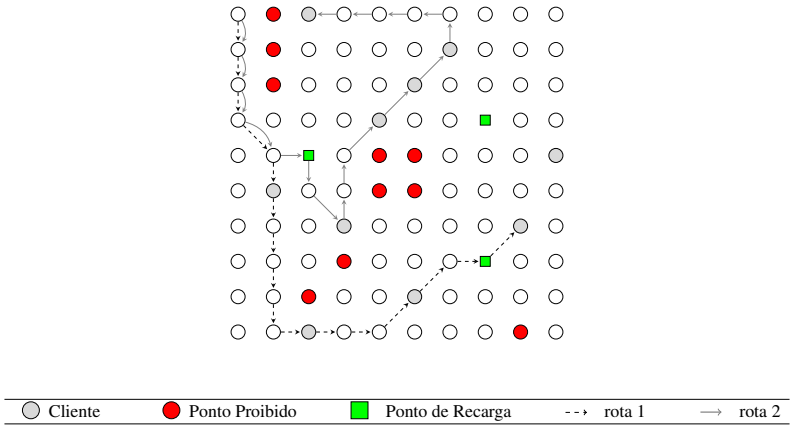


Figura 1. Exemplo de instância e solução do problema proposto

2.1. Objetivos

Como mencionado anteriormente, este artigo aborda um problema multiobjetivo, no qual queremos encontrar um conjunto de soluções não dominadas. Em um problema real de roteamento de VANTs, uma grande quantidade de variáveis deve ser considerada para se obter a solução mais adequada. No entanto, para tornar este problema real em um problema computável, mapeamos três objetivos principais que resumem, de forma satisfatória, como o sistema real deve se comportar.

Consumo: é desejável que durante o percurso, o veículo consuma o mínimo possível de bateria/combustível.

Carga final: é interessante terminar a rota com a máxima taxa de carga possível, garantindo que o drone esteja preparado para uma rota futura.

Tempo: o percurso total deve ser realizado no menor tempo possível.

Como pode ser visto, o algoritmo proposto neste artigo foca em encontrar um *trade-off* equilibrado entre soluções, uma vez que um elemento pode afetar outro. Quanto menor o tempo, maior a velocidade, logo, maior o consumo. Maior a carga final significa mais tempo gasto recarregando/abastecendo, o que resulta em tempos maiores.

2.2. Restrições

Para ter uma solução válida, a rota deve atender a alguns requisitos listados abaixo:

Consumo: o nível de combustível/bateria do veículo não deve ficar abaixo de zero em nenhuma parte do trajeto, isso significaria que o VANT ficaria sem combustível/energia no meio do trajeto. No entanto, se chegar a zero e a rota terminar ou o drone atingir um ponto de energia, isso não afeta a validade da solução.

Área proibida: na vida real, existem áreas onde os drones não podem acessar ou atravessar. Esta situação foi representada por pontos especiais espalhados pela rede. Se a rota contiver esses pontos, a solução é inválida.

2.3. Variáveis

Existem duas variáveis que afetam o resultado final das funções objetivo: velocidade no trecho e tempo de recarga.

Velocidade no trecho: influencia não apenas o tempo total do trajeto, mas também o consumo, pois quanto maior a velocidade (v), maior o consumo. O nível de combustível/bateria no final do trecho (f) é resultado do nível de combustível/bateria no início do trecho (f_0) diminuído pelo consumo fixo (c_f) e a velocidade multiplicada pelo coeficiente de consumo variável (c_v) conforme mostrado na equação 1.

$$f = f_0 - v \times c_v - c_f \tag{1}$$

Tempo na estação de : o tempo gasto na estação de energia (r) é somado ao tempo total gasto na rota. No entanto, se o veículo passar mais tempo nele, poderá acumular mais combustível/energia em sua bateria. O nível de combustível/bateria no final do trecho é resultado do nível de combustível/bateria no início do trecho aumentado pela quantidade de combustível/energia recarregada (f_r) conforme mostrado na equação 2.

$$f = f_0 + f_r \tag{2}$$

A equação 3 mostra que o tempo no final do trecho (t) é resultado do tempo no início do trecho (t_0) acrescido da quantidade de combustível/energia recarregada multiplicada pelo coeficiente de tempo por combustível/energia (t_f) conforme mostrado na equação 3.

$$t = t_0 + f_r \times t_f \tag{3}$$

3. Metodologia

A metaheurística GRASP com MOVND (G-MOVND) como busca local foi a escolhida para ser aplicada. Ela pode ser vista como uma metaheurística multi-start para problemas de otimização combinatória, em que cada iteração consiste basicamente em duas fases: construção e busca local.

3.1. Construção

Nesta fase do algoritmo proposto, um critério guloso de escolher iterativamente o cliente mais próximo da real posição é utilizado, seguindo a filsofia da fase 1 o GRASP [Resende and Ribeiro 2019].

3.2. Busca Local VND

Na busca local, utilizamos um algoritmo do tipo Variable Neighborhood Descent (VND), desenvolvido por N. Mladenović e E. Hansen [Mladenović and Hansen 1997] em 1997. Esta metaheurística funciona aplicando um método de busca local referente a uma estrutura de vizinhança para uma solução inicial x. Se a solução x' obtida for melhor que a anterior, atribuímos x' a x (x := x'), e continuamos a busca com a estrutura de vizinhança atual; caso contrário, nós o alteramos.

A variante multiobjetivo do VND (MOVND) [Duarte et al. 2015] foi implementada com 7 estruturas de vizinhança. Diferentes vizinhanças afetam diferentes funções objetivo. Como estávamos lidando com um conjunto de soluções e não apenas uma única solução final desejada, não usamos apenas uma melhor que é modificada através

das iterações, mas um conjunto de soluções. Uma solução inicial é gerada pelo GRASP, que agora passa por buscas VND exaustivas, as soluções vizinhas obtidas são inseridas em um conjunto global de soluções se não forem dominadas.

As vizinhanças são apresentadas a seguir, na ordem de execução da metaheurística:

- Swap: Esta vizinhança é responsável por trocar as posições dos clientes na rota. Se a solução gerada não for dominada ou igual a qualquer outra rota no conjunto de soluções atual, a nova solução é adicionada a ele.

- Remove Ponto de Recarga: Esta vizinhança é responsável por remover possíveis pontos de recarga em cada subcaminho (caminho entre clientes) e verificar se melhora a solução atual. A ideia é que retirando esse trecho, o VANT chegaria ao final da rota em menos tempo. Encontramos um caminho que liga dois clientes ou a origem a um cliente e o removemos da rota. Depois, vinculamos diretamente os dois clientes. Dessa forma, se houvesse um ponto de recarga neste subcaminho, ele seria removido.

- Recarga mais próxima: Após removermos pontos de recarga desnecessários, tentamos adicionar outros que possam melhorar a solução atual.

- Remover Repetidos: Após todas as operações anteriores terem sido executadas na rota, podemos ter inserido clientes repetidos na rota. A ideia desta vizinhança é retirar os repetidos, tentando diminuir o tamanho da rota e, consequentemente, diminuir o consumo e/ou tempo.

- Aumento/Diminuição de Velocidade na Seção: Aumenta/diminui a velocidade em 1 unidade em cada trecho inteiro da rota que liga dois pontos importantes (clientes e/ou ponto de recarga).

- Aumento/Redução Aleatório(a) de Velocidade: Aumenta/reduz a velocidade em 1 unidade em cada segmento de um subcaminho escolhido aleatoriamente.

- Aumento/Redução Aleatório(a) da Recarga: Aumenta/reduz em 1 unidade a porcentagem de carga em cada ponto de recarga de um subcaminho escolhido aleatoriamente.

3.3. Critério de Aceitação

Após gerar um vizinho da solução atual, a rota atual é avaliada e comparada com as rotas do conjunto de soluções. Será inserido se não for dominado por nenhum outro presente no conjunto de soluções. Se a nova rota dominar qualquer outra, esta última é removida do conjunto de soluções.

Neste caso de estudo, limitamos o tamanho do conjunto de soluções não dominadas, limitando, consequentemente, o número de operações (busca local). Se o conjunto estiver cheio e a nova rota não for dominada, ela só será inserida se houver uma solução no conjunto com resultado menor que a nova. Assim, a solução de entrada deve dominar, pelo menos, uma solução do conjunto atual.

3.4. Implementação VND

Para resolver o problema proposto gerando as rotas, diferentes versões de um algoritmo foram implementadas em C++. Foram implementadas quatro versões do método SWAP, por ser a vizinhança mais cara computacionalmente. As versões foram denominadas S1, S2, S3 e S4. O tamanho do conjunto de soluções também foi considerado como variável e diferentes testes foram feitos para verificar seu impacto nos resultados. Foram considerados os tamanhos 5, 15 e 30.

O MOVND foi a metaheurística escolhida, conforme abordado em sessões anteriores. No entanto, uma abordagem diferente também foi considerada. O MOVND original com um conjunto de soluções é conhecido por executar um loop para cada solução e dentro de um loop para cada vizinhança. Em nossa implementação atual, há uma inversão onde o loop externo percorre as vizinhanças e o interno percorre as soluções. Essa metaheurística foi denominada I-MOVND.

Algorithm 1 I-MOVND

```
1:  procedure I-MOVND(E, Vizinhança)
2:      repeat
3:          x ← Selecione(E \ Si)                ▷ Seleção entre os pontos não explorados
4:          for all k ∈ Vizinhança do
5:              inserido ← verdadeiro
6:              while inserido do
7:                  inserido ← false
8:                  x' ← vizinho(x, k)
9:                  if MO-Improvement(E, x, x') then
10:                     inserido ← falso
11:                     E ← Atualiza(E, x')
12:                 end if
13:             end while
14:         end for
15:     until E \ Si = ∅
16:     return E
17: end procedure
```

A atividade de calcular rotas pode ser muito custosa quando executada muitas vezes através do algoritmo. Para tanto, foi implementado um pré-processamento a fim de fazer comparações se trouxer ganhos aos resultados. O método consiste em pré-calcular as melhores rotas entre os pontos importantes do mapa (clientes, área proibida e pontos de recarga). As rotas são salvas e depois lidas como parte da entrada do problema.

Para resumir as diferentes implementações, cada amostra varia conforme:

- Instância
  - eil51A e eil51B: ambas com 51 clientes, 5% de pontos proibidos e 1% de pontos de recarga. A diferença entre elas se deve à posição dos pontos aleatórios proibidos e de recarga gerados.
  - eil101A e eil101B: igual ao anterior mas com 101 clientes.
- SWAP
  - S1: swap 1
  - S2: swap 2

  - S2-3: swap 2 e 3
  - S2-4: swap 2 e 4
  - S4: swap 4
- Busca Local
  - MOVND
  - I-MOVND
- Pré-processamento
  - Sim
  - Não

4. Experimentos computacionais

Como, pelo nosso conhecimento, não existe uma biblioteca de instâncias bem estabelecida para o problema abordado neste artigo, com todas as restrições aqui consideradas na literatura, foram criadas instâncias baseadas em duas bem conhecidas no formato 2D euclidiano foram usadas como base dos testes: Christofides/Eilon eil51 e eil101. Essas instâncias têm 51 e 101 clientes, respectivamente. A partir destas, foram gerados arrays representando a área que compreende todos os clientes, onde o valor de x varia do valor mínimo de x da instância ao máximo. O mesmo acontece com as coordenadas y.

Após a geração da matriz, os pontos nela contidos são escolhidos aleatoriamente e definidos como pontos de recarga e proibidos. Para fins de testes, foi estipulada uma taxa de 5% da matriz para pontos proibidos e 1% para pontos de recarga. Para cada instância, dois mapas diferentes foram gerados.

Cada amostra foi executada 5 vezes, cada uma por 5 minutos. Foi considerado o conjunto de soluções não dominadas de todas as 5 execuções em questão de resultados.

4.1. Medidas de Comparação

Para comparar os resultados obtidos durante os experimentos, foram utilizadas três medidas referentes à qualidade das soluções: hipervolume, cobertura e cardinalidade.

De acordo com [Talbi 2009], hipervolume é um indicador associado a uma aproximação dada pelo volume da porção do espaço objetivo que é fracamente dominada por um conjunto. Este indicador necessita da especificação de um ponto de referência Z que denote um limite superior sobre todos os objetivos. Neste problema foram utilizadas funções objetivo normalizadas. Além disso, assegurou-se que quanto mais próximo de 1 melhor é a medida. O código do cálculo do hipervolume é fornecido por [Fonseca et al. 2006].

As outras medidas são a cobertura e a cardinalidade. Cobertura, a quantidade percentual de soluções geradas por um método específico que está na referência de Pareto. Por exemplo, se quisermos comparar os métodos A e B, executaríamos os dois métodos e, no final, reuniríamos as soluções de ambos os métodos e selecionaríamos os únicos não dominados. Neste conjunto de soluções final, temos 4 soluções do método A e 6 soluções do método B. Então, a cobertura de A é de 40% e 60% de B.

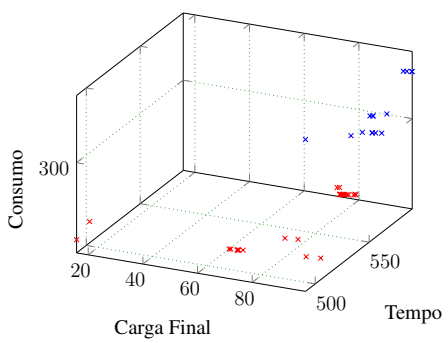
Cardinalidade é referente à quantidade absoluta.

4.2. Resultados Computacionais

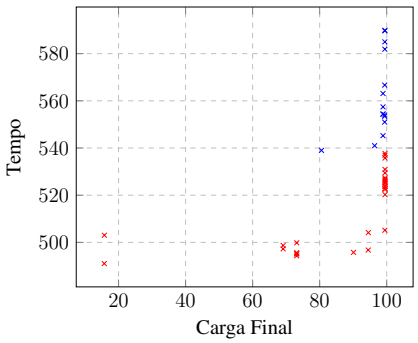
Os primeiros testes foram desenvolvidos para verificar os métodos que retornam os melhores resultados. Para tanto, foram utilizadas amostras com pré-processamento e conjunto de soluções de tamanho fixo de 30.

Os melhores resultados variaram entre S2-4 e S4 de VND e I-VND. Portanto, o próximo passo foi executar o algoritmo para avaliar como o tamanho do conjunto de soluções e o pré-processamento afetaram o resultado.

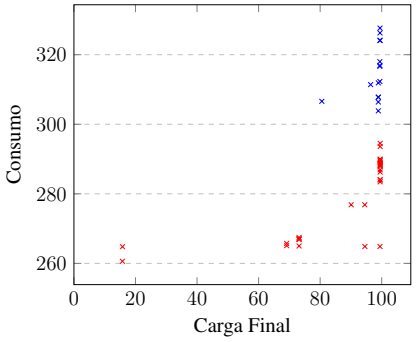
O pré-processamento resultou em melhores valores de hipervolume, cobertura e cardinalidade em quase todos os testes. Ou seja, o pré-cálculo de rotas gera um conjunto de soluções mais diversificado o que é um bom resultado quando se trata de um problema multiobjetivo. Em termos de tamanho máximo do conjunto, nas amostras com instância menor, quanto maior o conjunto de soluções, melhores os resultados. E com a instância maior, essa diferença não foi tão evidente, pois o tamanho da instância já ordena mais tempo computacional e um conjunto maior faz o mesmo.



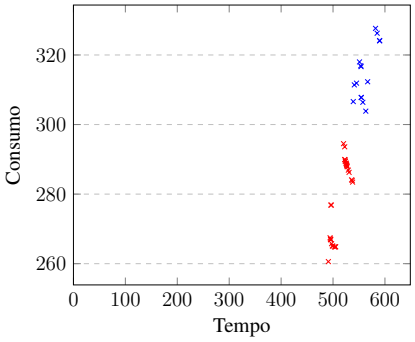
(a) Instância eil101 70% de bateria e 50% de clientes visitados.



(b) Carga Final x Tempo: VND-15 (azul) e VND-30 (vermelho)



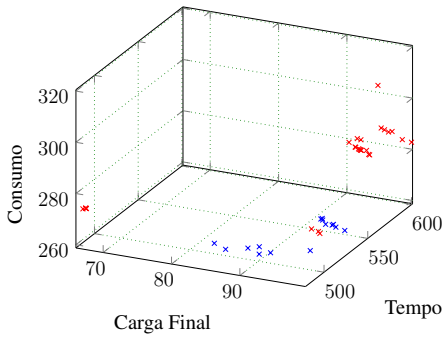
(c) Carga Final x Consumo: VND-15 (azul) e VND-30 (vermelho)



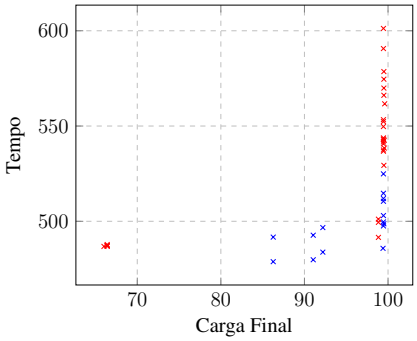
(d) Tempo x Consumo: VND-15 (azul) e VND-30 (vermelho)

Figura 2. Frentes de Pareto da Instância eil101-70-50

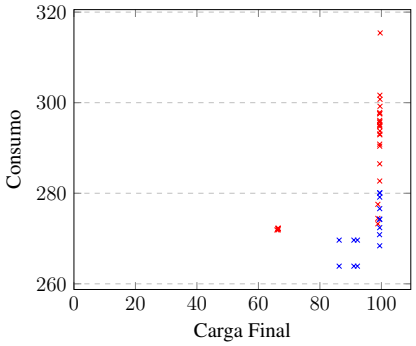
Nas Figuras 2 e 3 podemos ver as frentes de pareto geradas pela execução do algoritmo. Podemos inferir disso que o conjunto de soluções de tamanho 30 gera melhores resultados para instâncias maiores, enquanto o conjunto de tamanho máximo 15, gera melhores resultados com instâncias menores (menos pontos de inspeção/clientes), conjuntos menores.



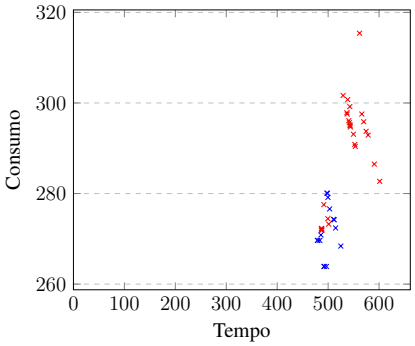
(a) Instância eil51 80% de bateria e 10% de clientes visitados.



(b) Carga Final x Tempo: VND-15 (azul) e VND-30 (vermelho)



(c) Carga Final x Consumo: VND-15 (azul) e VND-30 (vermelho)



(d) Tempo x Consumo: VND-15 (azul) e VND-30 (vermelho)

Figura 3. Frentes de Pareto da Instância eil51-80-10

5. Conclusões

A abordagem multiobjetivo e em grid, restrição de ancoragem, e a preocupação com o consumo (Green Computing) e o dinamismo deste problema mostram uma abordagem bastante prática, para aplicações reais. As instâncias consideraram posições iniciais arbitrárias de drones, e também capacidades iniciais variáveis de bateria, de modo que é possível integrar esta ferramenta em um solver online que resolva uma série de instâncias considerando mudanças nas características dinâmicas do cenário (devido às condições do vento, logística e outras restrições operacionais).

Este trabalho também mostra um potencial de crescimento aproximando-se cada vez mais de uma situação real. Assim, visualizamos trabalhos futuros levando em consideração mais camadas de rede representando o movimento vertical do drone. Além disso, um problema com drones heterogêneos e pontos proibidos temporários são algumas outras variáveis que podem ser exploradas em trabalhos futuros envolvendo esse problema.

Agradecimentos

Vitor N. Coelho agradece à agência brasileira FAPERJ (E-26/202.868/2016). Luiz S. Ochi foi apoiado pela FAPERJ e CNPq (301593/2013-2), Igor M. Coelho pela FAPERJ. Este estudo foi financiado em parte pela Coordenaçãoção de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) - Código de Finanças 001.

Referências

Adabo, G. J. (2014). Long range unmanned aircraft system for power line inspection of brazilian electrical system. *Journal of Energy and Power Engineering*, 8(2).

Agatz, N., Bouman, P., and Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, 52(4):965–981.

Coelho, B. N., Coelho, V. N., Coelho, I. M., Ochi, L. S., Zuidema, D., Lima, M. S., da Costa, A. R., et al. (2017). A multi-objective green uav routing problem. *Computers & Operations Research*, 88:306–315.

Coelho, V. N., Grasas, A., Ramalhinho, H., Coelho, I. M., Souza, M. J., and Cruz, R. C. (2016). An ils-based algorithm to solve a large-scale real heterogeneous fleet vrp with multi-trips and docking constraints. *European Journal of Operational Research*, 250(2):367–376.

de Sousa, M. M., González, P. H., Ochi, L. S., and de Lima Martins, S. (2021). A hybrid iterated local search heuristic for the traveling salesperson problem with hotel selection. *Computers & Operations Research*, 129:105229.

Deng, C., Wang, S., Huang, Z., Tan, Z., and Liu, J. (2014). Unmanned aerial vehicles for power line inspection: A cooperative way in platforms and communications. *J. Commun*, 9(9):687–692.

Duarte, A., Pantrigo, J. J., Pardo, E. G., and Mladenovic, N. (2015). Multi-objective variable neighborhood search: an application to combinatorial optimization problems. *Journal of Global Optimization*, 63(3):515–536.

Erdoğan, S. and Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114.

Floreano, D. and Wood, R. J. (2015). Science, technology and the future of small autonomous drones. *Nature*, 521(7553):460.

Fonseca, C. M., Paquete, L., and López-Ibánez, M. (2006). An improved dimension-sweep algorithm for the hypervolume indicator. In *2006 IEEE international conference on evolutionary computation*, pages 1157–1163. IEEE.

Gutin, G. and Punnen, A. P. (2006). *The traveling salesman problem and its variations*, volume 12. Springer Science & Business Media.

Haala, N., Cramer, M., Weimer, F., and Trittler, M. (2011). Performance test on uav-based photogrammetric data collection. *Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(1/C22):7–12.

Harris, A., Sluss, J. J., Refai, H. H., and LoPresti, P. G. (2005). Alignment and tracking of a free-space optical communications link to a uav. In *Digital Avionics Systems Conference, 2005. DASC 2005. The 24th*, volume 1, pages 1–C. IEEE.

Irizarry, J., Gheisari, M., and Walker, B. N. (2012). Usability assessment of drone technology as safety inspection tools. *Journal of Information Technology in Construction (ITcon)*, 17(12):194–212.

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European journal of operational research*, 59(3):345–358.

Máthé, K. and Buşoniu, L. (2015). Vision and control for uavs: A survey of general methods and of inexpensive platforms for infrastructure inspection. *Sensors*, 15(7):14887–14916.

Metni, N. and Hamel, T. (2007). A uav for bridge inspection: Visual servoing control law with orientation limits. *Automation in construction*, 17(1):3–10.

Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers & operations research*, 24(11):1097–1100.

Nigam, N. and Kroo, I. (2008). Persistent surveillance using multiple unmanned air vehicles. In *Aerospace Conference, 2008 IEEE*, pages 1–14. IEEE.

Resende, M. G. and Ribeiro, C. C. (2019). Greedy randomized adaptive search procedures: Advances and extensions. In *Handbook of metaheuristics*, pages 169–220. Springer.

Schermer, D., Moeini, M., and Wendt, O. (2018). A variable neighborhood search algorithm for solving the vehicle routing problem with drones. Technical report, Technical Report Technische Universität Kaiserslautern.

Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons.

Vansteenwegen, P., Souffriau, W., and Sörensen, K. (2012). The travelling salesperson problem with hotel selection. *Journal of the Operational Research Society*, 63(2):207–217.

Wang, X., Poikonen, S., and Golden, B. (2017). The vehicle routing problem with drones: several worst-case results. *Optimization Letters*, 11(4):679–697.