

Problema de Roteamento de Veículos com Tempo de Bateria e Demandas Baseadas em Distância

Cauã Magalhães Pereira

Julho 2024

Abstract

O problema de roteamento de veículos (VRP) com restrição de tempo de operação é um desafio crescente em diversas áreas, como logística e inspeções com drones. Este trabalho propõe um modelo de Programação Linear Inteira Mista (MILP) para resolver essa variante do VRP, considerando a velocidade dos veículos e o tempo de bateria como restrições principais. Utilizamos duas estratégias de eliminação de subcircuitos para garantir a viabilidade das soluções. Ao minimizar o tempo máximo de operação, otimizamos a utilização da frota e melhoramos a eficiência operacional. Os resultados obtidos podem ser aplicados em diversos setores, contribuindo para a otimização de rotas e a redução de custos.

Introdução

O problema de roteamento de veículos (VRP) é um clássico da otimização, com aplicações em diversas áreas. Uma variante relevante é o VRP com restrição de tempo de operação, motivado pelo crescente uso de drones em inspeções e entregas. A limitação do tempo de voo dos drones torna fundamental a otimização das rotas para garantir a conclusão das tarefas dentro do tempo de bateria disponível.

Neste trabalho, propomos um modelo de Programação Linear Inteira Mista (MILP) para o VRP com restrição de tempo de operação, considerando a velocidade dos veículos e o tempo de bateria como restrições principais. O objetivo é minimizar o tempo máximo de operação, garantindo que todos os pontos sejam atendidos dentro do tempo de voo disponível. Além disso, exploramos duas estratégias de eliminação de subcircuitos para garantir a viabilidade das soluções.

A contribuição deste trabalho reside em apresentar uma abordagem mais realista para o problema de roteamento de drones, considerando as especificidades da tecnologia. Os resultados obtidos podem ser aplicados em diversas áreas, como inspeção de linhas de transmissão, monitoramento ambiental e entregas urbanas.

Modelagem Matemática

A modelagem matemática do problema de roteamento de veículos (VRP) com a capacidade definida pelo tempo de bateria e demandas baseadas na distância percorrida é apresentada a seguir. Utilizamos variáveis binárias para representar as decisões de roteamento e variáveis contínuas para o tempo de operação dos veículos.

1.1 Definições e Notações

- V : Conjunto de todos os pontos, incluindo o depósito.
- K : Conjunto de veículos.
- $d_{i,j}$: Distância entre os pontos i e j .
- s_k : Velocidade do veículo k .
- b_k : Capacidade máxima do veículo k em termos de tempo de bateria.
- c_k : Capacidade de cobertura do veículo $k = b_k * s_k * 60$.
- S : Subconjunto de pontos visitados em V que formam um subcircuito.

1.2 Branch and Cut

1.2.1 Variáveis de Decisão

- **Variáveis binárias** $x_{i,j,k}$: Determinam se o veículo k viaja do ponto i ao ponto j .

$$x_{i,j,k} = \begin{cases} 1 & \text{se o veículo } k \text{ viaja do ponto } i \text{ ao ponto } j, \\ 0 & \text{caso contrário.} \end{cases}$$

- **Variáveis binárias** $y_{i,k}$: Determinam se o ponto i é visitado pelo veículo k .

$$y_{i,k} = \begin{cases} 1 & \text{se o ponto } i \text{ é visitado pelo veículo } k, \\ 0 & \text{caso contrário.} \end{cases}$$

- **Variável contínua** max_time:

$$\text{max_time} = \max \left(\sum_{i \in V} \sum_{j \in V, j \neq i} \frac{d_{i,j} \cdot x_{i,j,k}}{s_k} \right), \quad \forall k \in K$$

1.2.2 Função Objetivo

Minimizar o tempo máximo de cobertura de todos os pontos:

$$\text{minimize}(\text{max_time})$$

1.2.3 Restrições

- Cada ponto deve ser visitado exatamente uma vez por algum veículo, exceto o depósito:

$$\sum_{j \in V, j \neq i} \sum_{k \in K} x_{i,j,k} = \sum_{j \in V, j \neq i} \sum_{k \in K} x_{j,i,k} = \begin{cases} |K| & \text{se } i = 0, \\ 1 & \text{se } i \neq 0. \end{cases}$$

- Se um veículo visita um ponto, então ele deve sair desse ponto:

$$\begin{aligned} \sum_{j \in V, j \neq i} x_{i,j,k} &= y_{i,k}, \quad \forall i \in V, \forall k \in K, \\ \sum_{j \in V, j \neq i} x_{j,i,k} &= y_{i,k}, \quad \forall i \in V, \forall k \in K. \end{aligned}$$

- Tempo de viagem não pode exceder a capacidade máxima do veículo e o tempo máximo:

$$\begin{aligned} \sum_{i \in V} \sum_{j \in V, j \neq i} d_{i,j} \cdot x_{i,j,k} &\leq c_k, \quad \forall k \in K, \\ \sum_{i \in V} \sum_{j \in V, j \neq i} \frac{d_{i,j} \cdot x_{i,j,k}}{s_k} &\leq \text{max_time}, \quad \forall k \in K. \end{aligned}$$

- Eliminação de subcircuito:

$$y_{hk} \leq \sum_{i \in S} \sum_{j \in S^c} x_{ijk} \quad \forall h \in S, \forall k \in V \quad (1)$$

1.2.4 Resolução

Para resolver o problema, diferente da solução direta (Branch and Bound), utilizamos um método iterativo. Essa técnica é utilizada para resolver problemas de roteamento de veículos (VRP) que garante que a solução final não contenha subcircuitos (caminhos fechados menores que o circuito principal que não passam pelo depósito). A seguir, apresentamos um resumo de como esse método funciona:

- **Modelagem Inicial:** Define-se um modelo de programação linear inteira mista (MILP) para o VRP, incluindo variáveis de decisão, função objetivo e restrições.
- **Resolução Inicial:** O modelo é resolvido utilizando um solver (por exemplo, GLPK). Nesta etapa, é possível que a solução contenha subcircuitos.
- **Identificação de Subcircuitos:** Após a resolução inicial, a solução é analisada para identificar subcircuitos. Isso é feito construindo-se um grafo direcionado a partir das variáveis de decisão que representam as rotas dos veículos e buscando componentes fortemente conectadas (subcircuitos).
- **Eliminação de Subcircuitos:** Para cada subcircuito identificado, adiciona-se uma nova restrição ao modelo, forçando que pelo menos uma aresta (rota) do subcircuito seja rompida. Essa restrição é chamada de restrição de eliminação de subtour.

- **Iteração:** O modelo com as novas restrições é resolvido novamente utilizando o solver. O processo de identificação e eliminação de subcircuitos é repetido até que não existam mais subcircuitos na solução.
- **Convergência:** O método iterativo termina quando uma solução válida, sem subcircuitos, é encontrada. Esta solução representa a rota otimizada para todos os veículos, respeitando as restrições de capacidade (tempo de bateria) e demandas (distância).

1.3 Formulação MTZ

1.3.1 Variáveis de Decisão

- **Variáveis binárias $x_{i,j,k}$:** Determinam se o veículo k viaja do ponto i ao ponto j .

$$x_{i,j,k} = \begin{cases} 1 & \text{se o veículo } k \text{ viaja do ponto } i \text{ ao ponto } j, \\ 0 & \text{caso contrário.} \end{cases}$$

- **Variáveis binárias $y_{i,k}$:** Determinam se o ponto i é visitado pelo veículo k .

$$y_{i,k} = \begin{cases} 1 & \text{se o ponto } i \text{ é visitado pelo veículo } k, \\ 0 & \text{caso contrário.} \end{cases}$$

- **Variável contínua max_time :**

$$\text{max_time} = \max \left(\sum_{i \in V} \sum_{j \in V, j \neq i} \frac{d_{i,j} \cdot x_{i,j,k}}{s_k} \right), \quad \forall k \in K$$

- **Variáveis contínuas u_i :** Variáveis auxiliares para eliminação de subcircuitos.

1.3.2 Função Objetivo

Minimizar o tempo máximo de cobertura de todos os pontos:

$$\text{minimize} (\text{max_time})$$

1.3.3 Restrições

- Cada ponto deve ser visitado exatamente uma vez por algum veículo, exceto o depósito:

$$\sum_{j \in V, j \neq i} \sum_{k \in K} x_{i,j,k} = \sum_{j \in V, j \neq i} \sum_{k \in K} x_{j,i,k} = \begin{cases} |K| & \text{se } i = 0, \\ 1 & \text{se } i \neq 0. \end{cases}$$

- Se um veículo visita um ponto, então ele deve sair desse ponto:

$$\begin{aligned} \sum_{j \in V, j \neq i} x_{i,j,k} &= y_{i,k}, \quad \forall i \in V, \forall k \in K, \\ \sum_{j \in V, j \neq i} x_{j,i,k} &= y_{i,k}, \quad \forall i \in V, \forall k \in K. \end{aligned}$$

- Tempo de viagem não pode exceder a capacidade máxima do veículo e o tempo máximo:

$$\sum_{i \in V} \sum_{j \in V, j \neq i} d_{i,j} \cdot x_{i,j,k} \leq c_k, \quad \forall k \in K,$$

$$\sum_{i \in V} \sum_{j \in V, j \neq i} \frac{d_{i,j} \cdot x_{i,j,k}}{s_k} \leq \text{max_time}, \quad \forall k \in K.$$

- Eliminação de subcircuitos (MTZ):

$$u_j \geq u_i + (n - 1) \cdot x_{i,j,k} + (n - 3) \cdot x_{j,i,k} - (n - 2),$$

$$\forall i \in V, i \neq 0,$$

$$\forall j \in V, j \neq 0, i \neq j,$$

$$\forall k \in K$$

$$u_i \geq 1, \quad \forall i \in V, i \neq 0,$$

$$u_i \leq n - 1, \quad \forall i \in V, i \neq 0,$$

$$u_0 = 0.$$

1.3.4 Resolução

A resolução do problema utilizando a formulação MTZ é relativamente direta, uma vez que as restrições de Miller-Tucker-Zemlin são incorporadas diretamente ao modelo matemático. Assim, o solver pode aplicar um método de branch and bound tradicional para encontrar a solução ótima. Diferentemente do método de branch and cut 1.2.4, que envolve um processo iterativo de geração de planos de corte para eliminar subcircuitos, a formulação MTZ evita essa etapa, simplificando a resolução.

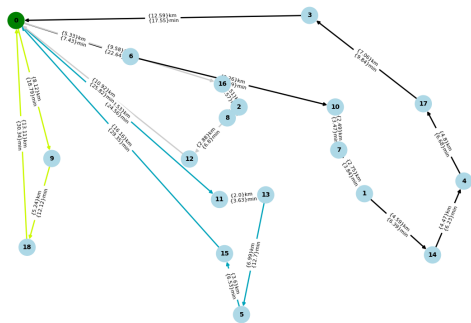
Resultados

A Tabela 1 apresenta os resultados obtidos ao comparar o método de planos de cortes com a formulação MTZ para resolver o problema de roteamento. Em termos de tempo de execução, os métodos se mostraram muito próximos apesar do MTZ aparentar ser mais consistente, mesmo com alguns outliers - ver teste 4.2 na tabela 1 -, com uma leve vantagem para o método de cortes em instancias menores.

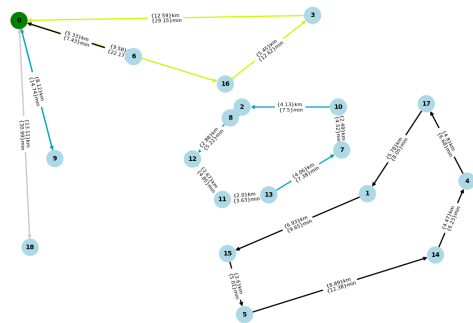
Um ponto extremamente notável é que, quando o algoritmo de cortes é interrompido pelo tempo limite, as soluções obtidas por esse método podem ser inviáveis (figura 1), ou seja, podem conter subcircuitos que não incluem o ponto de partida. Isso ocorre porque as restrições que eliminam os subcircuitos são adicionadas dinamicamente durante o processo de resolução, e o tempo limite pode impedir que todas as restrições necessárias sejam geradas.

A formulação MTZ, por sua vez, garante a viabilidade da solução encontrada, mesmo que esta não seja a solução ótima. Isso se deve às restrições específicas da formulação, que impedem a formação de subcircuitos. Contudo, essa garantia de viabilidade pode resultar em um tempo de execução maior em comparação com o método de planos de cortes.

Em resumo, a escolha entre o método de planos de cortes e a formulação MTZ depende das características específicas da aplicação. Se a prioridade for obter uma solução rapidamente, mesmo que ela não seja a melhor possível, o método de planos de cortes pode ser uma boa opção. Por outro lado, se a viabilidade da solução for um requisito fundamental, a formulação MTZ é mais adequada.



MTZ



Planos de Corte

Figure 1: Comparação soluções intermediárias

ID	NumVertices	NumCarros	MTZ Time(s)	CUTS Time(s)
1_1	11	5	503.8197	950.6443
1_2	11	4	4.563338	10.94854
2_1	14	4	1800.350 (tle)	1970.222 (tle)
2_2	14	3	1800.214 (tle)	1258.299
3_1	5	4	0.023806	0.0223908
3_2	5	3	0.0226922	0.0214853
4_1	15	3	1800.325	1926.184 (optimal)
4_2	15	2	244.6456	14.81131
5_1	14	4	4189.309 (tle)	4650.953 (tle)
5_2	14	3	3211.422 (tle)	3220.653 (tle)
6_1	12	5	601.8521	3990.829 (tle)
6_2	12	4	261.3638	157.5534
7_1	14	3	3730.173 (tle)	3828.658 (tle)
7_2	14	2	20.72704	7.789416
8_1	4	3	0.0341952	0.0200565
8_2	4	2	0.0206747	0.0185368
9_1	5	2	0.0225112	0.0428286
9_2	5	2	0.0244858	0.0696850
10_1	8	4	0.326361	1.986708
10_2	8	3	0.325610	1.263551
11_1	8	3	0.329739	0.4811406
11_2	8	2	0.0646813	0.2284524
12_1	10	5	11.88957	26.55118
12_2	10	4	27.96690	39.86329
13_1	11	2	0.325404	0.2538640
13_2	11	2	0.631650	1.984571
14_1	8	4	1.431984	5.315332
14_2	8	3	0.324672	0.8963320
15_1	7	4	0.0651164	0.0440297
15_2	7	3	0.107201	0.3346004
16_1	12	5	458.7638	2027.072 (tle)
16_2	12	4	255.3510	321.9270
17_1	15	3	1800.276 (tle)	1898.929 (optimal)
17_2	15	2	1800.297 (tle)	18.18392
18_1	10	4	34.95760	89.14834
18_2	10	3	2.149441	5.563742
19_1	5	3	0.0493305	0.0455136
19_2	5	2	0.0468450	0.1323206
20_1	6	3	0.0820906	0.1032033
20_2	6	2	0.0495598	0.1188879

Table 1: Tabela de Resultados

Referências Bibliográficas

- [1] Towards Data Science. (2020). The Vehicle Routing Problem: Exact and Heuristic Solutions. *Towards Data Science*. Disponível em: <https://towardsdatascience.com/the-vehicle-routing-problem-exact-and-heuristic-solutions-c411c0f4d734>.
- [2] L. Bertazzi, B. Golden, X. Wang. (2014). *Min-Max vs. Min-Sum Vehicle Routing: A worst-case analysis*. European Journal of Operational Research, 236(3), 822-830. Available at: <https://www.sciencedirect.com/science/article/pii/S0377221714005918>.