

Soluções factíveis para o problema do caixeiro viajante com veículo elétrico e janelas de tempo

Pedro Belin Castellucci, Álvaro Junio Pereira Franco, Rafael de Santiago¹

¹Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC)
Cx.P. 476 – 88040-370 Florianópolis – SC – Brasil

{pedro.castellucci, alvaro.junio, r.santiago}@inf.ufsc.br

Abstract. *Recently, electric vehicle routing problems have been explored in the optimization literature more and more. However, the electric traveling salesman problem with time windows remains one of the less explored variants. One challenge of this variant is in generating initial solutions for branch-and-cut solvers. To this end, we propose a simple and effective local search heuristic. The heuristic was able to find feasible solutions for 49 (of 50) benchmark instances.*

Resumo. *Problemas de roteamento com veículos elétricos têm ganhado atenção na literatura de otimização. No entanto, uma variante menos estudada é o problema do caixeiro viajante com veículos elétricos e janelas de tempo. Um dos desafios dessa variante está na obtenção de soluções iniciais factíveis para serem utilizadas em resolvedores baseados em branch-and-cut. Para geração dessas soluções, este trabalho propõe uma heurística de busca local simples e eficaz. A heurística foi capaz de gerar soluções factíveis para 49 (das 50) instâncias utilizadas na avaliação.*

1. Introdução

Problemas de roteamento de veículos (VRPs) são tradicionais na literatura de otimização [Laporte 2009]. Um dos motivos de sua importância prática é a possibilidade de extensões do problema básico para contemplar diversas características práticas como janelas de tempo, frota heterogênea, múltiplos depósitos, coleta e entrega de produtos entre outros [Lahyani et al. 2015].

Uma extensão de problemas de roteamento de veículos que vem recebendo interesse é o roteamento de veículos elétricos [Kucukoglu et al. 2021] (ECVRP). O principal desafio adicional dessa variante é a necessidade de acomodar características de veículos elétricos como a autonomia limitada, diferentes tecnologias de (re)carga de bateria e a visita a estações de recarga durante o trajeto.

Neste trabalho, explora-se uma versão particular do problema de roteamento de veículos elétricos: o caixeiro viajante com veículo elétrico e janelas de tempo (ETSPTW) [Roberti e Wen 2016]. Além da importância prática, o ETSPTW também é importante para métodos de decomposição para versões do ECVRP, como métodos de geração de colunas por exemplo.

No melhor de nosso conhecimento, embora diversas características de roteamento de veículos elétricos já tenham sido estudadas, o ETSPTW continua carente

de investigações. Para o caso do roteamento de uma frota de veículos, dentre as características já estudadas estão diferentes tecnologia e modelos para recarga de bateria [Felipe et al. 2014, Montoya et al. 2017, Keskin e Çatay 2018], estações para trocas de bateria [Wang et al. 2023], inclinação nas estradas [Rastani 2020] e velocidade variável do veículo [Abdallah e Adel 2020]. Para o caso de caixeiro viajante (apenas um veículo sem restrição de capacidade), destacam-se os trabalhos [Doppstadt et al. 2016, Doppstadt et al. 2020], que consideram apenas um veículo com diferentes modos de operação (puramente elétrico e puramente à combustão, entre outros). Por fim, um trabalho que estuda o ETSPTW é [Erdoğdu e Karabulut 2021], no entanto, os autores consideram que a carga de bateria inicial é suficiente para visitar todos os clientes.

Um trabalho que explora o ETSPTW da mesma forma que estamos explorando é o de [Roberti e Wen 2016]. Os autores propõem dois modelos: um modelo de tamanho polinomial; e um exponencial em relação ao tamanho da entrada. Um dos desafios para o uso do modelo polinomial é a dificuldade em se encontrar soluções iniciais. O modelo exponencial, por sua vez, só é viável sob algumas hipóteses, que limitam sua aplicabilidade. Atacando esse desafio, a contribuição deste trabalho está no desenvolvimento de uma heurística capaz de fornecer soluções factíveis, que podem ser utilizadas como soluções iniciais para resolvedores e métodos exatos, com o objetivo de resolver, de forma exata, instâncias maiores.

2. Descrição do problema

Seja $V = \{o\} \cup C \cup S$ um conjunto de vértices em que o é o vértice de origem e destino de um *tour*, C um conjunto de consumidores e S um conjunto de estações de recarga. Além disso, uma janela de tempo $[e_i, l_i]$, $e_i, l_i \in \mathbb{Z}_+$ é associada a cada vértice $i \in C$ para definir o intervalo em que cada consumidor deve ser atendido. É permitido esperar caso o tempo de chegada seja inferior a e_i .

Seja $A = V \times V$ um conjunto de arcos possíveis de serem percorridos. A cada arco (i, j) está associada uma distância $d_{ij} > 0$, um tempo de viagem $t_{ij} > 0$ e um consumo de bateria $q_{ij} = h \cdot d_{ij} > 0$, em que h é a taxa de consumo de bateria. A capacidade da bateria é denotada por $Q > 0$. Assume-se que o tempo de recarregamento em uma estação é proporcional à carga faltante de bateria, com uma taxa constante g , igual para todas as estações. Além disso, é permitido que múltiplas recargas sejam feitas na mesma estação. Por fim, assume-se que, ao visitar uma estação de recarga, o veículo recarrega totalmente a bateria. O objetivo é encontrar um *tour*, com distância (custo) mínima, respeitando a capacidade de bateria e as janelas de tempo. Uma modelagem de programação inteira pode ser consultada em [Roberti e Wen 2016].

3. Proposta de heurística

Um dos desafios para o modelo CF proposto por [Roberti e Wen 2016] está na dificuldade em encontrar soluções factíveis, quando utilizados resolvedores baseados em *branch-and-cut*. Por isso, propomos uma busca local baseada na vizinhança 2-opt, frequentemente utilizada em problemas de roteamento [Zhang et al. 2022] e elemento da heurística estado-da-arte, para o problema do caixeiro viajante [Helsgaun 2000].

A vizinhança 2-opt é utilizada em uma busca local (Algoritmo 1) que iterativamente encontra o melhor par de arestas para serem removidas (linha 4) para gerar uma

nova solução. Então, se necessário, atualiza-se a melhor solução encontrada até o momento (linha 6). Na sequência, verifica-se se a solução é factível. Em caso afirmativo, encerra-se o algoritmo (linha 8). No caso de a solução da linha 4 ter o mesmo valor da melhor solução atual, então o algoritmo se encerra (linha 11) tendo encontrado um ótimo local.

Algoritmo 1: Busca local proposta

```

1 Function local-search (solution, max_iter) :
2   best = solution
3   for i = 1, ..., max_iter do
4     candidate = f2opt(best)
5     if value(candidate) < value(best) then
6       best = candidate
7       if value(best) = 0 then
8         return best
9     else
10      if value(candidate) = value(best) then
11        return; // Ótimo local.

```

A medida de qualidade de solução (função *value*) é definida por penalizações por violação de carga máxima de bateria disponível e de janelas de tempo $[e_i, \ell_i], i \in C$. Para cada trecho sem carregamento na rota, se verifica a quantidade de bateria do veículo ao chegar em uma estação de recarga (ou no destino final), q_i . Caso $q_i < 0$, soma-se $|q_i|$ à penalidade relacionada à bateria. De forma análoga, seja τ_i o horário de chegada do veículo no consumidor i , soma-se $\max\{\tau_i - \ell_i, 0\}$ à penalidade relacionada às janelas de tempo. A penalidade total da rota é dada por (1).

$$value(rota) = \alpha \sum_{i \in C} \max\{\tau_i - \ell_i, 0\} + \beta \sum_{i \in C: q_i < 0} |q_i|. \quad (1)$$

4. Resultados

O algoritmo de busca local foi implementado em Python e executado em um computador com 8GB de memória RAM, Intel® Core™ i5-8265U CPU@1.6GHZ com sistema operacional Linux Mint 21. Para avaliação, foram utilizadas as instâncias de [Roberti e Wen 2016]. As instâncias são agrupadas em relação ao número de consumidores, duração de janelas de tempo e número de estações de recarga. Para cada instância, se gerou uma permutação aleatória dos conjuntos de consumidores e estações de recarga para aplicar a busca local. Em caso de infactibilidade da solução, o algoritmo era reiniciado com uma nova permutação aleatória para outra tentativa (foram realizadas, no máximo, 200 tentativas). Nos experimentos foram utilizados $\alpha = \beta = 1$.

A heurística de busca local é eficaz em achar soluções factíveis em média e mediana. Em 49 (de 50) instâncias, a busca local encontrou soluções factíveis. Em média, foram necessários menos de 10 tentativas para 5 (das 10) classes de instâncias (Tabela 1). Em mediana, foram encontradas soluções factíveis com menos de 10 tentativas para todas as classes. O número de movimentos 2-opt necessários para cada tentativa bem sucedida não ultrapassou 32 (em média).

Tabela 1. Média e mediana do número de tentativas para cada grupo de instância, do número de movimentos 2-opt realizados na tentativa bem sucedida, o custo da solução, o gap ($\frac{custo - z^*}{z^*}$) e tempo computacional (em segundos). Em que z^* é a melhor solução encontrada por [Roberti e Wen 2016] e cada grupo contém cinco instâncias.

Classe		Tentativas	Mov.	Custo	Gap	t (s)
n20w120s5	Média	29.80	21.20	334.00	21.10	12.73
	Mediana	8.00	21.00	350.00	15.14	3.16
n20w120s10	Média	47.20	29.00	338.25	29.50	46.31
	Mediana	8.00	32.00	355.50	29.47	7.63
n20w140s5	Média	13.00	20.80	312.20	31.72	5.64
	Mediana	7.00	17.00	324.00	36.13	2.52
n20w140s10	Média	9.60	22.00	327.60	40.34	8.46
	Mediana	5.00	23.00	337.00	41.92	5.17
n20w160s5	Média	17.20	17.00	309.00	36.21	7.09
	Mediana	3.00	18.00	307.00	35.77	0.98
n20w160s10	Média	11.20	20.60	341.80	52.91	9.10
	Mediana	3.00	21.00	355.00	52.02	2.49
n20w180s5	Média	4.00	18.60	322.00	32.46	1.30
	Mediana	5.00	18.00	319.00	33.01	1.44
n20w180s10	Média	5.00	23.80	352.00	48.30	3.69
	Mediana	2.00	24.00	360.00	41.73	2.00
n20w200s5	Média	4.00	15.20	324.60	30.22	1.20
	Mediana	3.00	14.00	320.00	32.58	0.80
n20w200s10	Média	1.60	22.80	350.80	43.28	1.23
	Mediana	1.00	25.00	370.00	48.83	1.04

Para mais de metade das instâncias o tempo computacional para encontrar uma solução viável é menor do que 2 segundos. Ao se agrupá-las em classes, o tempo computacional mediano de cada classe não ultrapassou 10 segundos. Por outro lado, o valor (custo) da solução factível encontrada é pelo menos 15% maior do que a solução ótima da respectiva instância, podendo chegar a valores superiores a 50%.

5. Conclusões e trabalhos futuros

O problema do caixeiro viajante com veículo elétrico e janelas de tempo (ETSPTW) é uma variante NP-difícil relativamente recente de problemas de roteamento. Apesar de sua importância prática e também no contexto de métodos de decomposição, ainda há uma carência de métodos de solução para tal variante.

Nesse trabalho, foi proposta uma heurística para auxiliar na obtenção de soluções factíveis iniciais. A heurística é simples e eficiente, sendo capaz de obter soluções factíveis para 49 (das 50) instâncias utilizadas para avaliação. Como trabalhos futuros, pretende-se avaliar o comportamento dos pacotes computacionais quando fornecidas tais soluções e utilizá-las como solução de partida para métodos exatos visando resolver instâncias maiores (com mais de 20 clientes).

Agradecimentos

Este trabalho foi parcialmente suportado pela Conselho Nacional de Desenvolvimento Científico e Tecnológico (405247/2023-0) e pela Universidade Federal de Santa Catarina e pela Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina.

Referências

- Abdallah, K. S. e Adel, Y. (2020). Electric vehicles routing problem with variable speed and time windows. *Industry, Engineering & Conference Management Systems Conference*.
- Doppstadt, C., Koberstein, A., e Vigo, D. (2016). The hybrid electric vehicle–traveling salesman problem. *European Journal of Operational Research*, 253(3):825–842.
- Doppstadt, C., Koberstein, A., e Vigo, D. (2020). The hybrid electric vehicle—traveling salesman problem with time windows. *European Journal of Operational Research*, 284(2):675–692.
- Erdoğdu, K. e Karabulut, K. (2021). A new model for minimizing the electric vehicle battery capacity in electric travelling salesman problem with time windows. *Turkish Journal of Electrical Engineering and Computer Sciences*, 29(5):2545–2560.
- Felipe, Á., Ortuño, M. T., Righini, G., e Tirado, G. (2014). A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review*, 71:111–128.
- Helsgaun, K. (2000). An effective implementation of the lin–kernighan traveling salesman heuristic. *European journal of operational research*, 126(1):106–130.
- Keskin, M. e Çatay, B. (2018). A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. *Computers & operations research*, 100:172–188.
- Kucukoglu, I., Dewil, R., e Cattrysse, D. (2021). The electric vehicle routing problem and its variations: A literature review. *Computers & Industrial Engineering*, 161:107650.
- Lahyani, R., Khemakhem, M., e Semet, F. (2015). Rich vehicle routing problems: From a taxonomy to a definition. *European Journal of Operational Research*, 241(1):1–14.
- Laporte, G. (2009). Fifty years of vehicle routing. *Transportation science*, 43(4):408–416.
- Montoya, A., Guéret, C., Mendoza, J. E., e Villegas, J. G. (2017). The electric vehicle routing problem with nonlinear charging function. *Transportation Research Part B: Methodological*, 103:87–110.
- Rastani, S. (2020). *Route planning of electric freight vehicles by considering internal and environmental conditions*. PhD thesis.
- Roberti, R. e Wen, M. (2016). The electric traveling salesman problem with time windows. *Transportation Research Part E: Logistics and Transportation Review*, 89:32–52.
- Wang, D., Ding, A., Chen, G., e Zhang, L. (2023). A combined genetic algorithm and A* search algorithm for the electric vehicle routing problem with time windows. *Advances in Production Engineering & Management*, 18(4).
- Zhang, H., Ge, H., Yang, J., e Tong, Y. (2022). Review of vehicle routing problems: Models, classification and solving algorithms. *Archives of Computational Methods in Engineering*, pages 1–27.