

Soluções Híbridas de Geração de Colunas e Algoritmo Genético para o Problema de Escalonamento de Veículos Elétricos com Múltiplos Depósitos

Leonardo Soares da Silva

Universidade Federal do Rio Grande do Sul

Escola de Administração – Rua Washington Luiz, 855 - Centro Histórico, Porto Alegre – RS

leonardo.soares@ufrgs.br

Pablo Cristini Guedes

Universidade Federal do Rio Grande do Sul

Escola de Administração – Rua Washington Luiz, 855 - Centro Histórico, Porto Alegre – RS

pablo.guedes@ufrgs.br

RESUMO

Nos últimos anos, o uso de Veículos Elétricos (EVs) cresceu significativamente diante da necessidade de reduzir emissões de gases de efeito estufa. Isso motiva o estudo do Problema de Escalonamento de Veículos Elétricos (EVSP), que busca otimizar a operação de frotas elétricas. Este trabalho propõe investigar a variante Multi-Depot do EVSP (MD-EVSP), que considera múltiplas garagens com capacidades limitadas. O objetivo é comparar três critérios: eficiência (tempos de processamento dos algoritmos), eficácia (valores finais das funções objetivo) e qualidade (ocorrência de violações de restrições operacionais). Serão implementadas e testadas abordagens híbridas: AG-GC, que aplica um Algoritmo Genético para gerar soluções iniciais seguido de Geração de Colunas; e GC-AG, que inverte essa ordem. Ambas as técnicas serão avaliadas em instâncias adaptadas de estudos anteriores. Espera-se que os achados contribuam para a otimização econômica e operacional de frotas elétricas em cenários urbanos e avancem estado da arte em escalonamento de veículos.

PALAVRAS CHAVE. Algoritmo Genético, Geração de Colunas, Veículos Elétricos.

Tópicos (indique, em ordem de PRIORIDADE, o(s) tópico(s) de seu artigo)

ABSTRACT

In recent years, the use of Electric Vehicles (EVs) has grown significantly due to the need to reduce greenhouse gas emissions. This trend motivates the study of the Electric Vehicle Scheduling Problem (EVSP), which aims to optimize the operation of electric fleets. This dissertation proposes to investigate a complex variant, the Multi-Depot EVSP (MD-EVSP), which considers multiple depots with limited capacities. The objective is to compare three criteria: efficiency (algorithm processing times), effectiveness (final objective-function values), and solution quality (occurrence of operational-constraint violations). Two hybrid approaches will be implemented and tested: AG-GC, which applies a Genetic Algorithm to generate initial solutions followed by Column Generation; and GC-AG, which reverses that order. Both techniques will be evaluated on instances adapted from prior studies. The findings are expected to contribute to the economic and operational optimization of electric fleets in urban environments and to advance the state of the art in vehicle scheduling.

KEYWORDS. Genetic Algorithm. Column Generation. Electric Vehicles.

Paper topics (indicate in order of PRIORITY the paper topic(s))

1. Introdução

A eletrificação do transporte tem ganhado crescente relevância frente aos desafios ambientais e energéticos. Em 2023, as vendas globais de veículos elétricos (EVs) atingiram quase 14 milhões de unidades, representando 18% do total de automóveis vendidos no mundo, com forte concentração nos mercados da China, Europa e Estados Unidos [International Energy Agency 2024]. Projeções indicam que, até 2035, metade do comércio de veículos deverá ser composta por EVs, o que poderá evitar a demanda por aproximadamente 10 milhões de barris de petróleo por dia.

No Brasil, apesar do crescimento no número de EVs, a adoção em larga escala ainda enfrenta obstáculos, especialmente no transporte público. Em junho de 2022, havia 351 ônibus elétricos em operação no país [Ministério do Desenvolvimento Regional; Banco Mundial 2022], concentrados em cidades como São Paulo, Curitiba e Campinas. A escassez de infraestrutura de recarga e a complexidade operacional da eletrificação do transporte coletivo tornam urgente o desenvolvimento de estratégias de planejamento eficientes. Nesse contexto, o escalonamento de frotas de EVs surge como um desafio essencial, visando a sustentabilidade ambiental e a otimização econômica e operacional do transporte de passageiros e mercadorias [Pasha et al. 2024].

O problema de escalonamento de veículos elétricos (EVSP) trata da alocação de veículos a viagens predefinidas, respeitando restrições operacionais e buscando minimizar custos. A natureza elétrica dos veículos adiciona complexidade significativa ao problema devido às restrições de autonomia limitada, tempos de recarga, disponibilidade de carregadores e capacidade da rede elétrica, tornando-o NP-difícil mesmo em casos simples [Sassi and Oulamara 2014]. Sua variante com múltiplos depósitos (MD-EVSP) contribui para a complexidade ao considerar alocação de veículos em diferentes garagens com capacidades limitadas. Este trabalho compara duas abordagens híbridas para o MD-EVSP: AG-GC, onde um Algoritmo Genético gera soluções iniciais que são refinadas por Geração de Colunas; e GC-AG, onde o GC antecipa o AG, e este último é utilizado para seleção das "melhores" rotas finais para composição da solução.

2. Trabalhos Relacionados

Diversas abordagens têm sido propostas para a resolução do *Electric Vehicle Scheduling Problem* (EVSP) e de problemas relacionados ao roteamento e escalonamento de veículos. Neste contexto, destacam-se os métodos de Geração de Colunas (GC), os Algoritmos Genéticos (AGs) e, mais recentemente, abordagens híbridas que combinam ambas as técnicas.

A GC tem sido amplamente utilizada na resolução de problemas de escalonamento e roteamento devido à sua capacidade de lidar com grandes instâncias de forma estruturada. Trabalhos seminais como [Ribeiro and Soumis 1994] e [Feillet et al. 2004] demonstraram o uso da GC como técnica exata para subproblemas como o *Elementary Shortest Path Problem with Resource Constraints* (ESPPRC), frequentemente adotado em problemas de escalonamento. Abordagens posteriores têm buscado acelerar a convergência. Por exemplo, [Pepin et al. 2009] propuseram o GC truncado, no qual múltiplas colunas são adicionadas por iteração e a execução é interrompida com base em critérios de melhoria marginal. A combinação desse método com heurísticas de *large neighborhood search* mostrou resultados competitivos na resolução do *Multi-Depot Vehicle Scheduling Problem* (MDVSP). Estratégias de redução do espaço de busca, como em [Guedes and Borenstein 2015], também contribuíram para tornar o GC mais viável em contextos operacionais.

Em paralelo, AGs têm sido aplicados com sucesso em diversos problemas de roteamento e escalonamento, dada sua flexibilidade na representação de soluções e capacidade de evitar ótimos locais [Talbi 2009; Pereira and Tavares 2009]. No contexto do EVSP, [Wang et al. 2021b] utilizaram AGs para integrar decisões de roteamento e consumo energético, enquanto [Abid et al. 2024] propuseram uma abordagem multiobjetivo, com inserção posterior de estações de recarga. Tais estratégias demonstram a versatilidade dos AGs em modelagens realistas.

Particularmente relevantes para este trabalho são os estudos que combinam GC e AG em abordagens híbridas. [Wang et al. 2021a] propuseram uma solução GC-AG para o MD-EVSP utilizando heurísticas para geração de soluções iniciais, GC truncado para geração de colunas candidatas, e AG para selecionar as colunas finais via estratégia elitista. Apesar da redução significativa no tempo de processamento em comparação com o *branch-and-price*, os autores relataram a ocorrência de viagens não cobertas ou duplicadas. [Dunbar et al. 2020] utilizaram AG para resolver o subproblema de precificação no GC, enquanto [Zhou et al. 2024] aplicaram uma estratégia colaborativa de escalonamento e regulação de velocidade com GC e AG para frotas de ônibus autônomos, alcançando forte redução no tempo computacional.

Esses trabalhos evidenciam o valor da integração entre GC e AG, embora diferentes ordens de aplicação dessas técnicas possam impactar diretamente a qualidade e viabilidade das soluções. Isso motiva a investigação da abordagem inversa (AG-CG), proposta neste estudo.

3. Definição e Formulação do Problema

Sejam ST uma relação de viagens i a serem realizadas, com suas respectivas demandas d_i de passageiros e horários de partida e chegada pré-estabelecidos; S um conjunto de terminais de ônibus elétricos, em que cada $s \in S$ representa um terminal de partida ou chegada de uma ou mais viagens $i \in ST$; (3) K o conjunto de garagens de armazenamento dos veículos elétricos, com suas respectivas capacidades, em que as garagens de $o(k)$ de origem e $d(k)$ de destino são as mesmas; e C o conjunto de estações para recarga das baterias dos veículos. Uma solução do MD-EVSP objetiva minimizar o custo do escalonamento da frota de veículos de modo que todas as viagens sejam executadas, as capacidades dos depósitos sejam respeitadas, e nenhum veículo fique sem carga elétrica ao longo de um trajeto.

Na instância definida por [Wang et al. 2021a] (e parcialmente replicada neste trabalho), cada linha de ônibus possui um terminal de origem e um terminal de destino e cada terminal possui uma tabela de horários de partida. O roteiro de um veículo em um dia compreende sua saída de uma garagem, a realização de uma série de viagens (incluindo recargas de bateria), e o retorno para a mesma garagem de origem. A formulação desse tipo de problema como uma rede de conexão facilita sua modelagem e compatibilidade com os métodos de solução propostos, conforme evidenciado em [Wang et al. 2021a], [Guedes and Borenstein 2015] e [Pepin et al. 2009].

3.1. Rede de Conexão

A rede de conexão (CN) para o MD-EVSP, tal como definida por [Wang et al. 2021a], é representada por um grafo $G^k = (V^k, A^k)$ para cada garagem $k \in K$. Constituem os nós da rede: o conjunto de viagens ST , cada uma com horário de partida e chegada; o conjunto de garagens K , cada uma com sua respectiva capacidade máxima de veículos; o conjunto de terminais S , que são os pontos de partida e chegada das viagens; e o conjunto de estações de recarga C . Os arcos $a \in A^k$ representam deslocamentos entre duas viagens, entre terminais e depósitos, ou entre terminais e estações de recarga, e são denominados como arcos de *deadhead* por conta da ausência de passageiros. Sua definição é a que segue.

Seja $A_{pout} = \{(o(k), i) | k \in K, i \in V \setminus K\}$ o conjunto de arcos *deadhead* representando saídas dos veículos de suas respectivas garagens. Seja $A_{pin} = \{(i, d(k)) | i \in V \setminus K, k \in K\}$ o conjunto de arcos *deadhead* de reingresso dos veículos em suas respectivas garagens. Os demais arcos *deadhead* são dados por $A_{dh} = \{(i, j) | et_i + \delta_{ij} \leq st_j, i, j \in ST\} \cup \{(i, c) | i \in ST, c \in C\} \cup \{(c, j) | st_c + r + \delta_{cj} \leq st_j, c \in C, j \in ST\}$, em que et_i representa o horário de finalização da viagem i , δ_{ij} é o tempo de deslocamento entre um nó i e um nó j , st_j é o horário de início da viagem j , st_c representa o momento em que a recarga de bateria de um veículo é iniciada em uma estação de recarga c , r é o tempo de recarga ou troca da bateria de um veículo e δ_{cj} representa o tempo de deslocamento entre um nó c e um nó j . O conjunto de arcos da rede é expresso, então, como $A^k = A_{pout} \cup A_{pin} \cup A_{dh}$. Com isso, tem-se elementos suficientes para a formulação matemática.

3.2. Formulação Matemática

Valendo-se da CN e da notação já apresentadas, e também apoiando-se em [Pasha et al. 2024], [Guedes and Borenstein 2015] e [Pepin et al. 2009], o MD-EVSP pode ser escrito da seguinte forma (Problema Mestre — PM):

$$\text{PM: min } \sum_{k \in K} \sum_{(i,j) \in A^k} c_{ij} x_{ij}^k \quad (1)$$

$$\text{sujeito a } \sum_{k \in K} \sum_{j: (i,j) \in A^k} x_{ij}^k = 1 \quad \forall i \in ST \quad (2)$$

$$\sum_{j: (i,j), i \in ST, j \in C, (i,j) \in A_{dh}} x_{ij}^k \leq 1 \quad \forall i \in ST \quad (3)$$

$$\sum_{j: (o(k),j) \in A^k} x_{ij}^k \leq v^k \quad \forall k \in K \quad (4)$$

$$\sum_{j: (j,i) \in A^k} x_{ji}^k - \sum_{j: (i,j) \in A^k} x_{ij}^k = 0, \forall i \in V^k \setminus \{o(k), d(k)\}, \forall k \in K \quad (5)$$

$$y_j^k \leq y_i^k - (h \cdot \gamma_{ij}) x_{ij} + b(1 - x_{ij}), \forall (i,j) \in V^k, \forall k \in K \quad (6)$$

$$y_j^k \leq b - (h \cdot \gamma_{ij}) x_{ij}, \forall (i,j) \in V^k, \forall k \in K \quad (7)$$

$$x_{ij}^k \in \{0, 1\}, y_i \geq 0, \quad \forall (i,j) \in A^k, \forall k \in K. \quad (8)$$

A função objetivo (1) minimiza o custo total do escalonamento, em que c_{ij} é o custo de um arco (i, j) . O conjunto de equações (2) impõe que cada viagem i seja executada uma única vez. As inequações (3) impedem que, a partir de uma mesma viagem i , ocorra mais de um arco para uma estação de recarga. As restrições (4) limitam o número de veículos despachados, em que v é a capacidade de armazenamento de veículos de cada garagem. As equações (5) impõem a conservação de fluxo. As restrições (6) e (7) garantem que nenhum veículo fique sem bateria ao longo de quaisquer trajetos, onde y_j^k é o restante de bateria de um veículo ao chegar em um nó i ou j , h é a taxa de consumo de energia dos veículos por unidade de distância, e b é a capacidade total de bateria de cada veículo. As inequações (8) expressam os domínios das variáveis.

O custo total de um roteiro p dá-se por $c^p = c_p^b + c_p^r + c_p^o$, em que: c_p^b é o custo fixo de alocação de um veículo em um roteiro p ; c_p^r é o custo de recarga de um veículo ao longo de um roteiro p ; e c_p^o é o custo operacional do roteiro. O custo operacional, por sua vez, é composto por: (a) o custo de uma viagem, dado pela multiplicação da sua duração por c^t , que é o custo por unidade de tempo de um veículo transportando passageiros; (b) o custo de deslocamento de um veículo vazio, resultante da multiplicação entre a duração desse deslocamento e c^d , que é o custo por unidade de tempo de um veículo em um arco *deadhead*. Assim, tem-se que o custo operacional de um roteiro p para um veículo em um dia, representado por c_p^o , é dado pela soma entre os custos operacionais das viagens (a), dos arcos de *deadhead* (b), e c^w , que é o custo do veículo parado.

4. Método

O método proposto envolve o desenvolvimento, execução e comparação de dois processos para resolver o MD-EVTSP. O primeiro processo é composto pelas seguintes etapas: (1) uso de um método heurístico para gerar um banco de rotas p viáveis (2) aplicação do AG para geração de uma solução inicial aprimorada; e (3) utilização do método de Geração de Colunas (GC) truncado para encontrar a solução final do problema. Essa abordagem é denominada AG-GC, indicando que o AG é aplicado antes do GC.

O segundo processo consiste em uma inversão parcial da ordem das tarefas do primeiro método, resultando na seguinte sequência: (1) aplicação do mesmo método heurístico da etapa 1 do primeiro processo, porém gerando apenas uma única solução inicial; (2) aplicação do GC para aprimorar a solução através da inserção de novas colunas com custo reduzido negativo; e (3) uso do AG para selecionar as "melhores" colunas, ou seja, aquelas que têm o maior impacto negativo na função objetivo, compondo a solução final para o problema. Denomina-se essa abordagem como GC-AG, significando que o método de GC é utilizado anteriormente ao AG.

Ambas as técnicas serão avaliadas em uma mesma instância adaptada de [Wang et al. 2021a], e comparadas com base em três critérios principais: (i) tempo de processamento; (ii) valor da função objetivo final; e (iii) qualidade da solução, esta última medida pela ausência de viagens não cobertas ou duplicadas e pelo respeito às capacidades dos depósitos. Ainda que o método GC-GC esteja sendo replicado de [Wang et al. 2021a] (com algumas adaptações), até o momento da escrita deste trabalho não foram localizadas referências que houvessem aplicado a solução AG-GC ao problema MD-EVSP. Nas subseções seguintes, detalha-se os algoritmos e processos utilizados nas soluções.

4.1. Heurística para Geração de Soluções Iniciais

Esse método envolve uma busca iterativa por viagens a serem cobertas, e a subsequente atribuição de veículos para execução dessas viagens, considerando a capacidade dos depósitos e as necessidades de recarga de baterias. O procedimento é finalizado quando todas as viagens tiverem sido processadas. A cada iteração do algoritmo, é gerada uma rota p^k . A heurística proposta é uma adaptação, com refinamentos, da utilizada por [Wang et al. 2021a].

A Tabela (1) relaciona as variáveis e os parâmetros específicos utilizados no algoritmo para execução do método heurístico apresentado nesta seção. Os demais, não listados na tabela, compõem o modelo matemático e a rede de conexão, previamente definidos.

Tabela 1: Parâmetros do Método Heurístico

Símbolo	Descrição
$method$	Argumento que indica o método, podendo ser "GC-GA" ou "GA-GC".
$dist^k$	Distância acumulada por um veículo oriundo de $k \in K$ ou de uma estação de recarga
D	Autonomia total máxima, em distância, de um veículo
$time^k$	Tempo acumulado de operação de um veículo oriundo de um depósito $k \in K$
r_b	Tempo total até a recarga completa da bateria de um veículo
W	Tempo total máximo de expediente diário para quaisquer veículos
$covered$	Conjunto de viagens cobertas
$routes^k$	Conjunto de roteiros
$plan_{ij}^d$	Distância planejada entre nós i e j
$plan_{ij}^t$	Tempo planejado entre nós i e j

Com o método heurístico definido, o processo de geração de soluções iniciais segue um fluxo estruturado, expresso através do Algoritmo (1).

Algorithm 1 Método Heurístico para Geração de Soluções Iniciais

1 Atribuição de variáveis: Atribua $n \leftarrow 0$, $covered \leftarrow \emptyset$, $routes \leftarrow \emptyset$, $dist^k \leftarrow 0$ e $time^k \leftarrow 0$.

while $ST \not\subseteq covered$ **do**

2 Seleção da primeira viagem:

if $method = "GA-GC"$ **then**

2.1 Listagem top-10: Selecione as primeiras 10 viagens (ou um número menor caso hajam menos de 10) do conjunto $ST \in \setminus covered$, por ordem de horário de início (do menor ao maior), dentre todas as tabelas. Atribua à $top10$ o conjunto de viagens selecionadas.

2.2 Seleção aleatória: Selecione, aleatoriamente, do conjunto $top10$, uma única viagem. Atribua à variável i a viagem selecionada. Vá para o passo 3.

else

2.3 Seleção por ordem cronológica: Escolha, dentre todas as tabelas de horário, a viagem $i \in ST \setminus covered$ que acontece mais cedo. Em caso de empate, selecione a de menor duração. Se o empate persistir, selecione aleatoriamente uma dentre as empatadas.

3 Alocação de veículo: Escolha o depósito k mais próximo de ss_i com capacidade suficiente.

4 Inicialização de rota: Inicialize a rota p^k com o arco $(o(k), i)$. Atualize $dist^k$ e $time^k$. Atribua $covered \leftarrow covered \cup \{i\}$.

5 Busca por próxima viagem: Busque, aleatoriamente, uma viagem $j \notin covered$ tal que $et_i + \delta_{ij} \leq st_j$, e $plan_{ij}^d + plan_{jc}^d \leq D$.

if não encontrada OU $time^k + plan_{ij}^t > W$ **then**

5.1: Adicione arco $(i, d(k))$, finalize p^k , $routes \leftarrow routes \cup \{p^k\}$. Retorne ao passo 2.

if $D - dist^k < plan_{ij}^d$ **then**

5.2: Adicione arco (i, c) até estação de recarga mais próxima, atualize $dist^k \leftarrow 0$, $time^k \leftarrow time^k + \delta_{ic} + r_b$, e defina $i \leftarrow c$. Retorne ao passo 5.

6. Adição da viagem j à rota: Adicione j à rota p^k , atualize $dist^k$, $time^k$. Atribua $covered \leftarrow covered \cup \{j\}$. Defina $i \leftarrow j$. Retorne ao passo 5.

7. Retorne: $routes$

Dependendo do método utilizado, tem-se uma abordagem diferente na seleção da primeira viagem. Se o método for GA-GC, a condicional expressa no passo 2.1 do Algoritmo (1) faz, primeiramente, uma pré-seleção das dez primeiras viagens, em termos de horário de início, e somente uma dentre as pré-selecionadas é, então, sorteada para ser a primeira viagem, repetindo-se o processo a cada nova rota. Caso seja utilizado o método GC-AG, então o algoritmo segue a mesma estrutura de [Wang et al. 2021a] em relação à primeira seleção: a viagem que acontece mais cedo, dentre todas as tabelas de horário, é selecionada a cada nova rota gerada.

4.2. Algoritmo Genético Aplicado ao MD-EVSP

O AG proposto é flexível para compor as abordagens AG-GC e GC-AG. Na primeira, ele gera uma solução inicial que será refinada pelo GC; na segunda, atua após o GC, selecionando colunas e buscando a solução final. Para isso, a codificação de uma solução precisa ser versátil.

Uma solução do MD-EVSP consiste em rotas Ω^k , cada uma executada por um veículo de depósito $k \in K$. A rota é representada como uma sequência de nós $v \in V^{kf}$: viagens $i \in ST$, estações $c \in C$ e depósitos $o(k) = d(k)$. Cada par de nós define um arco implícito $a \in A^{kf}$. A codificação de uma solução assume uma sequência binária, onde cada gene representa a presença (1) ou ausência (0) de uma rota. Em contexto genético, essa sequência é o cromossomo de um indivíduo I_n . A Figura 1 ilustra esse conceito.

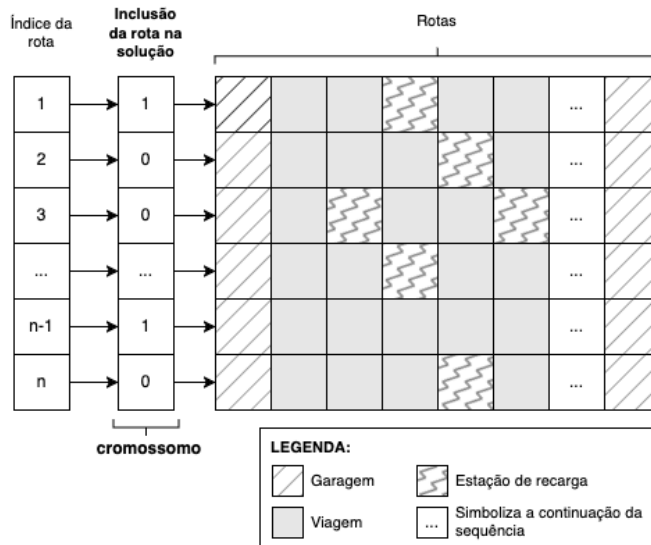


Figura 1: Exemplo de codificação de uma solução X .

O procedimento completo do GA está definido no Algoritmo 2, e a lista dos parâmetros nele utilizados consta na Tabela 2.

Tabela 2: Parâmetros e Variáveis do Algoritmo Genético

Símbolo	Descrição
p_{cross}	Probabilidade com que o operador de <i>crossover</i> é aplicado em pares de indivíduos.
p_{mut}	Probabilidade com que o operador de mutação é aplicado a genes de descendentes.
ω_0	Peso de penalização por violação da capacidade de depósitos.
ω_1	Peso do número de veículos utilizados na função de aptidão.
ω_2	Peso associado ao total de viagens repetidas ou não cobertas.
ω_3	Peso associado à presença de viagens muito curtas na solução.
ω_4	Constante no numerador da função de aptidão.
$MaxRoutes$	Número máximo de rotas (genes ativos) por indivíduo.
$MaxInd$	Número máximo de indivíduos por geração.
$MaxGen$	Número total de gerações no processo evolutivo.
I_n	Indivíduo (solução) da população, representado por um vetor binário $\mathbf{v}_n$.
$\mathbf{v}_n$	Vetor binário que indica presença (1) ou ausência (0) de cada rota na solução.
$\mathbf{u}_n$	Vetor de probabilidades de seleção de cada gene na geração de I_{n+1} .
P_n	População de indivíduos correspondente à geração n .
E	Lista de elite contendo os melhores indivíduos de gerações anteriores.
$offspring$	Conjunto de descendentes gerados por reprodução em cada geração.
I_{best}	Melhor indivíduo de uma dada geração ou da população final.

4.2.1. Inicialização do AG

Para gerar-se um indivíduo, um vetor $\mathbf{v}_0$ é inicializado com zeros. A cada iteração, uma rota viável ainda não utilizada é ativada ($g_i = 1$), priorizando aquelas cujo terminal de início $ss \in S$ ainda não foi coberto. Para isso, utiliza-se a lista L , que registra os terminais já atendidos por rotas ativadas, evitando repetições. Quando todas as rotas de um terminal são exauridas, ele é movido

para a lista L' , impedindo novas tentativas naquele terminal. Ao esgotar $L \cup L'$, L é reiniciada e o processo continua até atingir $MaxRoutes$ genes ativados.

A população inicial P_0 é composta por $MaxInd$ indivíduos gerados aleatoriamente. Cada indivíduo é formado por até $MaxRoutes$ rotas, selecionadas do conjunto Ω^k . A partir do primeiro indivíduo, gera-se um vetor de pesos u_0 . Trata-se de um vetor de pesos, de modo que $u_i = 0.8$ se $g_i = 1$, e $u_i = 0.2$ caso contrário. Em outras palavras, são atribuídas probabilidades maiores para a seleção de rotas não utilizadas pelo indivíduo anterior. Os próximos indivíduos seguem o mesmo processo, mas guiados por u_n , que é atualizado a cada iteração com base no indivíduo de índice $n - 1$. Assim, P_0 é construído de forma diversificada.

4.2.2. Processo de Evolução

A cada geração, indivíduos são selecionados por roleta, proporcional à aptidão. A reprodução aplica *crossover* e mutação para gerar descendentes. Os mais aptos têm maior chance de reprodução, e um mecanismo elitista garante a preservação de bons indivíduos, mantendo os melhores indivíduos em uma lista E , renovada a partir da terceira geração por uma roleta secundária. A função de aptidão ϕ avalia cada solução:

$$\phi = \frac{\omega_4}{\omega_1 v + \omega_2(r + u) + \omega_3 s} \quad (9)$$

onde v é o número de veículos; r e u o número de viagens repetidas e não cobertas; e s , as muito curtas. Para a abordagem AG-GC, adiciona-se penalização por violações de capacidade das garagens:

$$\phi_{(GA-GC)} = \frac{\omega_4}{\omega_0 h + \omega_1 v + \omega_2(r + u) + \omega_3 s} \quad (10)$$

onde h é o número de violações. O peso ω_1 corresponde ao custo de alocação de um veículo, e será definido com base em experimentação. Os pesos ω_2 , ω_3 e ω_4 têm como valores iniciais os sugeridos por [Wang et al. 2021a], respectivamente: 0, 6, 0, 4 e 1000).

A cada seleção, um operador de *crossover* divide o cromossomo em segmentos de 20 genes, trocando-os entre dois indivíduos com 50% de chance. Por sua vez, um operador de mutação altera aleatoriamente genes individuais. Ambos operadores ajudam a explorar o espaço de soluções. A aplicação do operador de mutação ocorre com 20% de chance.

O AG encerra após $MaxGen$ gerações. A melhor solução I_{best} da geração final é selecionada para uso nos métodos AG-GC ou GC-AG.

Algorithm 2 Algoritmo Genético para o MD-EVTSP

Passo 1: Inicialização

Atribua $gen \leftarrow 0$, $E \leftarrow \emptyset$, $P_0 \leftarrow \emptyset$
for $n = 0$ até $MaxInd - 1$ **do**

 Gere indivíduo I_n com até $MaxRoutes$ genes ativados, seguindo critério de diversificação

 Se $n = 0$: atribua $\mathbf{u}_0 \leftarrow$ vetor de pesos, com $u_i = 0.8$ se $g_i = 1$, e $u_i = 0.2$ caso contrário

 Se $n > 0$: gere I_n com base em $\mathbf{u}_{n-1}$ (probabilidades atualizadas)

 Atualize $\mathbf{u}_n$ com base em I_n . Atribua $P_0 \leftarrow P_0 \cup \{I_n\}$
Passo 2: Evolução
while $gen < MaxGen$ **do**

 2.1: Elitismo

 Identifique $I_{best} \in P_{gen}$ e atualize $E \leftarrow E \cup \{I_{best}\}$

 if $|E| > 3$ **then**

 Use roleta elitista proporcional à aptidão para remover um indivíduo de E

 2.2: Seleção e Reprodução

 Atribua $offspring \leftarrow \emptyset$

 while $|offspring| < MaxInd - |E|$ **do**

 Selecione dois pais I_a, I_b via roleta proporcional à aptidão

 Aplique operadores de *crossover* e mutação, com respectivas probabilidades p_{mut} e p_{cross} .

 Adicione descendente gerado a $offspring$

 2.3: Atualização da população

 Atribua $P_{gen+1} \leftarrow E \cup offspring$. Atribua $gen \leftarrow gen + 1$
Passo 3: Retorne I_{best} de P_{MaxGen} como melhor solução encontrada.

4.3. Geração de Colunas Aplicada ao MD-EVTSP

Conforme demonstrado por [Ribeiro and Soumis 1994] para o caso do MDVSP, o MD-EVSP também pode ser formulado como um Problema de Particionamento de Conjuntos. As abordagens de [Wang et al. 2021a], de [Guedes and Borenstein 2015], dentre outras, beneficiam-se dessa formulação em algum momento, por conta da utilização do GC.

Seja Ω^k o conjunto de todos os possíveis trajetos realizados por um veículo (colunas), tendo como origem uma garagem $o(k)$ e como destino final a mesma garagem, denotada por $d(k)$, com $k \in K$. Assim, $p \in \Omega^k$ é o roteiro de um veículo em um dia. Seja c_p o custo de execução de um roteiro p . Seja q a capacidade (lotação máxima) de um veículo. Seja a_{ip} a variável binária que sinaliza se a viagem $i \in V^k$ é coberta ($a_{ip} = 1$) ou não ($a_{ip} = 0$) pelo roteiro p . Seja θ_p a variável binária que denota se o roteiro p é incluído ($\theta_p = 1$) ou não ($\theta_p = 0$) na solução. A formulação do MD-EVSP como um Problema de Particionamento de Conjuntos é:

$$\min \sum_{k \in K} \sum_{p \in \Omega^k} c_p \theta_p \quad (11)$$

$$\text{sujeito a: } \sum_{k \in K} \sum_{p \in \Omega^k} a_{ip} \theta_p = 1, \quad \forall i \in ST \quad (12)$$

$$\sum_{p \in \Omega^k} \theta_p \leq v^k, \quad \forall k \in K \quad (13)$$

$$\theta_p, a_{ip} \in \{0, 1\}, \quad \forall p \in \Omega^k, \forall i \in T. \quad (14)$$

A função objetivo (11) minimiza o custo total. As restrições (12) garantem que cada viagem i seja coberta exatamente por um único roteiro p . A restrição (13) garante que a capacidade

máxima v de cada garagem seja respeitada. A restrição (14) delimita o domínio das variáveis de decisão. Estando o MD-EVSP modelado neste formato, cada rota viável de um veículo representa uma coluna no problema, propiciando a aplicação do método GC. Sejam α e β as variáveis duais correspondentes às restrições (2) e (4) do problema-mestre. Tem-se, então, o seguinte subproblema (*pricing problem* — PP), em que x_{ij}^k é a variável binária do problema primal:

$$(PP): \min \quad -\beta_k + \sum_{(i,j) \in A^k} (c_{ij} - \alpha_j) x_{ij}^k \quad (15)$$

$$\text{sujeito a:} \quad (3), (5) - (8) \quad (16)$$

Conforme observado por [Wang et al. 2021a], o PP apresentado trata-se do problema do caminho mais curto (*Shortest Path Problem*) com restrições de recursos, e o algoritmo de correção de rótulos *Shortest Path Faster Algorithm* (SPFA) é uma abordagem eficiente para solucioná-lo.

O algoritmo SPFA é utilizado para identificar-se a rota de menor custo, com arcos de peso negativo, a partir do nó de origem $o(k)$ até o nó de destino $d(k)$ na rede de conexão do problema MD-EVSP. Percorre-se a rede G^k para encontrar o caminho mais curto, considerando todos os nós em G como ponto de origem. Uma fila do tipo "primeiro a entrar, primeiro a sair" (FIFO — *first-in-first-out*) é utilizada para reduzir o número de nós a serem otimizados. No SPFA, uma rota p consiste em múltiplos nós pelos quais um veículo passa, e corresponde a um rótulo único denotado por $l_p = \{c_p, d_p, s\}$, em que c_p é o custo reduzido da rota p , d_p é a distância acumulada percorrida pelo veículo desde uma estação de recarga de baterias ou da garagem de origem do mesmo, e s é a sequência de nós na rota. O Algoritmo (3), adaptado de [Wang et al. 2021a], apresenta o procedimento de geração de novas colunas utilizando o SPFA para resolver o subproblema. A mesma técnica de truncagem utilizada pelos autores é aplicada.

Algorithm 3 Geração de Colunas com SPFA

Passo 1 (Inicialização): Um conjunto inicial de colunas $\Omega^k = \{p_1^k, p_2^k, \dots, p_n^k\} \mid k \in K, n \in \mathbb{N}^+$ é previamente fornecido. Atribua a Z um grande valor *big_M*. Seja *count* a variável que representa a contagem de iterações do GC. Atribua *count* $\leftarrow 0$.

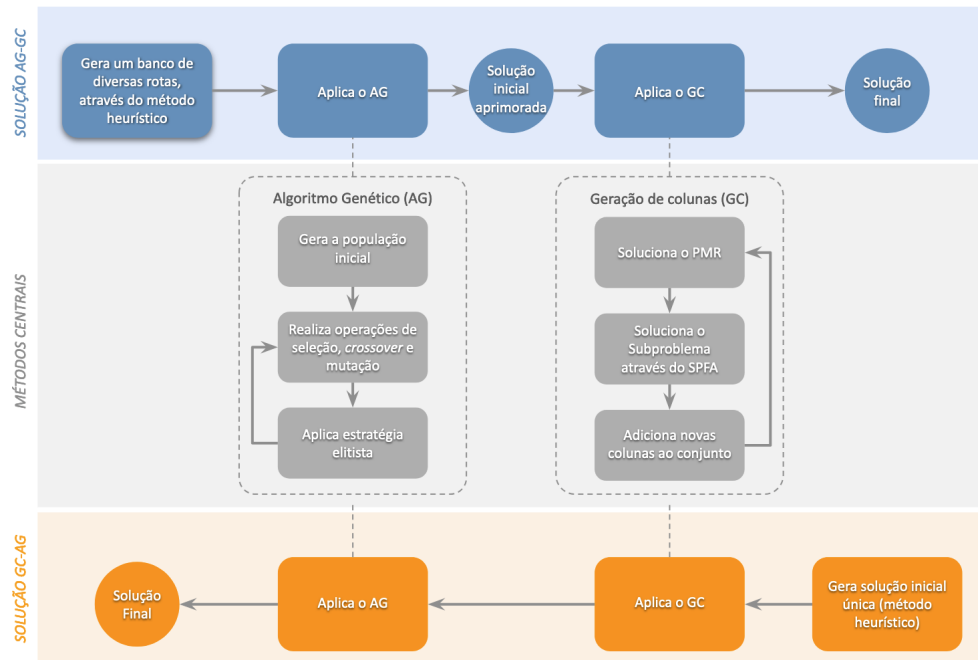
Passo 2 (Solução do PMR): O *software* CPLEX é utilizado para solução do PMR relaxado, utilizando Ω^k . Como resultado, obtêm-se as variáveis duais α e β . Atribua a Z' o valor da função objetivo do PMR solucionado. Se $Z - Z' < Z_{min}$, atribua *count* $\leftarrow count + 1$. Atribua $Z \leftarrow Z'$.

Passo 3 (Geração de Colunas): Construa a matriz de adjacência M , na qual cada elemento m_{ij} representa a distância entre o nó i e o nó j . Para cada nó $i \in ST$, utilize o SPFA para gerar um conjunto R_i de rotas (colunas) mais curtas, tendo i como ponto de partida, e garantindo que $dist^k \leq D$ e $time^k \leq W$. Para cada par de rotas $l_1 = \{\bar{c}_1, d_1, s_1\}$ e $l_2 = \{\bar{c}_2, d_2, s_2\}$ encontradas e adicionadas a R_i , se $\bar{c}_1 \leq \bar{c}_2$ e $d_1 \geq d_2$, a rota l_1 é melhor que a rota l_2 , pelo que a rota l_2 é removida de R_i . O conjunto de todas as rotas (colunas) geradas é dado por $R = \sum_{i \in ST} R_i$.

Passo 4 (Adição de Colunas): Ordene o conjunto R de colunas geradas em termos dos seus custos reduzidos c_p , de forma crescente. Cada coluna é representada por um rótulo $l_p = \{\bar{c}_p, d_p, s\}$. Selecione um subconjunto $R_{top} \subseteq R$ das primeiras K colunas de R após a ordenação. Para cada coluna em R_{top} : se $\bar{c}_p \geq 0$, a coluna p não é útil para obter-se uma solução aprimorada do problema mestre e, portanto, é descartada. Do contrário, a coluna p pode ser útil para encontrar-se uma solução melhor para o PMR e, portanto, é incluída no conjunto de colunas: $\Omega^k \leftarrow p$.

Passo 5 (Critério de Parada): Se *count* for maior que um dado número I , encerre o algoritmo. Do contrário, retorne ao Passo 2.

Figura 2: Diagrama das soluções propostas.



Fonte: O Autor.

A etapa de GC consiste na solução do Problema Mestre Restrito (PMR) expresso em (11)-(14) através do *software* CPLEX e, posteriormente, do seu respectivo PP, por sua vez resolvido utilizando-se o SPFA descrito no Algoritmo (3). A cada ciclo, a solução do PMR atualiza as variáveis duais α e β , utilizadas na solução do PP para geração de novas colunas a serem inseridas no PMR, iniciando uma nova iteração. A forma e o objetivo com os quais GC e AG (descrito na Seção (4.2)) são aplicados nas abordagens propostas difere de uma para a outra, conforme explicado a seguir.

4.4. Solução Através do AG-GC

Inicialmente, um banco Ω^k de rotas p^k distintas e viáveis (equivalentes a colunas do MD-EVTSP) é gerado, utilizando-se o método heurístico descrito na Seção (4.1), aplicado de forma iterada. Em seguida, o AG descrito na Seção 4.2 é empregado para alcançar uma solução inicial aprimorada, denotada por X'_0 , composta pelos melhores indivíduos p^k . Nesse caso, utiliza-se a função de aptidão específica para aplicação primeira do AG, dada pela Equação (10). Finalmente, a Geração de Colunas truncada é aplicada, segundo o Algoritmo (3), utilizando X'_0 como ponto de partida para a solução inicial, objetivando a obtenção da solução final para o problema. O segmento superior da Figura (2) resume a abordagem AG-GC.

4.5. Solução Através do GC-AG

Uma única solução inicial X_0 é gerada através do método heurístico descrito na Seção (4.1). Em seguida, a Geração de Colunas truncada é aplicada, conforme o Algoritmo (3) obtendo-se um conjunto de colunas com custo reduzido negativo. Por fim, o AG descrito na Seção 4.2 é empregado para se chegar à solução final, por meio da evolução da solução encontrada pelo GC, utilizando a Equação (9) como função de aptidão. O segmento inferior da Figura (2) resume a abordagem.

5. Resultados Preliminares

Ambas as abordagens citadas na seção estão sendo testadas em instâncias artificialmente geradas, adaptadas de [Wang et al. 2021a], e comparadas em termos de tempo de processamento,

valor das respectivas funções objetivo e qualidade das soluções (viagens não cobertas, viagens cobertas em duplicidade, violação de capacidades das garagens). A adaptação da instância decorre do fato de que não há informações suficientes sobre o padrão de espaçamento entre horários, bem como os horários de início e de fim, das tabelas de horário de cada terminal. Desta forma, delimitou-se horários de início e fim que podem ser inferidos dos parâmetros do problema, além de um intervalo para seleção aleatória de valores, com função de distribuição de probabilidades que mimetiza horários de pico.

Os experimentos estão sendo feitos utilizando-se a linguagem Python, em um computador com sistema operacional Windows, processador Intel i5 de 2,5 Giga-Hertz, com 8 Gigabytes de memória RAM. Até o momento, a modelagem do problema, o algoritmo de geração das soluções iniciais e o código de chamada ao solver CPLEX foram implementados. Os parâmetros utilizados são os mesmos de [Wang et al. 2021a]. A heurística de construção das soluções iniciais apresentou tempo de rodagem inferior 4 segundos, evidenciando que sua utilização em sequência é factível no método AG-GC. A solução do PMR em uma primeira iteração, através do CPLEX, ocorreu em menos de 2 segundos. O algoritmo de geração de colunas, contendo as soluções iteradas do PMR e a posterior sequência de aplicações do SPFA, apresentou tempo de rodagem de 35 minutos. Para o método GC-AG, o tempo total preliminar médio é de 45 minutos. Para o método GC-AG, o tempo total preliminar médio é de 51 minutos. Ao final do mês de Setembro deste ano, todos os experimentos terão sido desenvolvidos e validados, com potenciais mudanças nos resultados aqui apresentados, e informações completas acerca das funções objetivo e dos escalonamentos.

Referências

- Meryem Abid, Mohamed Tabaa, and Hanaa Hachimi. Electric vehicle routing problem with an enhanced vehicle dispatching approach considering real-life data. *Energies*, 17(7), 2024. ISSN 1996-1073. doi: 10.3390/en17071596. URL <https://www.mdpi.com/1996-1073/17/7/1596>.
- Michelle Dunbar, Simon Belieres, Nagesh Shukla, Mehrdad Amirghasemi, Pascal Perez, and Nishikant Mishra. A genetic column generation algorithm for sustainable spare part delivery: application to the sydney droppoint network. *Annals of Operations Research*, 290(1):923–941, July 2020. ISSN 1572-9338. doi: 10.1007/s10479-018-2911-2. URL <https://doi.org/10.1007/s10479-018-2911-2>.
- Dominique Feillet, Pierre DEJAX, Michel Gendreau, and Cyrille Gueguen. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44:216–229, 10 2004. doi: 10.1002/net.20033.
- Pablo C. Guedes and Denis Borenstein. Column generation based heuristic framework for the multiple-depot vehicle type scheduling problem. *Computers Industrial Engineering*, 90:361–370, 2015. ISSN 0360-8352. doi: <https://doi.org/10.1016/j.cie.2015.10.004>. URL <https://www.sciencedirect.com/science/article/pii/S0360835215003976>.
- International Energy Agency. Global ev outlook 2024, 2024. URL <https://www.iea.org/reports/global-ev-outlook-2024>.
- Ministério do Desenvolvimento Regional; Banco Mundial. Diagnóstico e benchmarking em Ônibus elétricos no atual contexto brasileiro. Technical report, Ministério do Desenvolvimento Regional, 2022. URL <https://www.gov.br/cidades/pt-br/central-de->

[conteudos/publicacoes/mobilidade-urbana/arquivos/diagnostico-e-benchmarking-em-onibus-eletricos-no-atual-contexto-brasileiro.pdf](#). Acesso em: 23 set. 2024.

Junayed Pasha, Bokang Li, Zeinab Elmi, Amir M. Fathollahi-Fard, Yui yip Lau, Abdolreza Roshani, Tomoya Kawasaki, and Maxim A. Dulebenets. Electric vehicle scheduling: State of the art, critical challenges, and future research opportunities. *Journal of Industrial Information Integration*, 38: 100561, 2024. ISSN 2452-414X. doi: <https://doi.org/10.1016/j.jii.2024.100561>. URL <https://www.sciencedirect.com/science/article/pii/S2452414X24000050>.

Ann-Sophie Pepin, Guy Desaulniers, Alain Hertz, and Dennis Huisman. A comparison of five heuristics for the multiple depot vehicle scheduling problem. *J. Scheduling*, 12:17–30, 02 2009. doi: 10.1007/s10951-008-0072-x.

Francisco Baptista Pereira and Jorge Tavares, editors. *Bio-inspired Algorithms for the Vehicle Routing Problem*. Studies in Computational Intelligence. Springer Berlin, Heidelberg, 1 edition, 2009. ISBN 978-3-540-85151-6. doi: 10.1007/978-3-540-85152-3. Published: 10 September 2008.

Celso C. Ribeiro and François Soumis. A column generation approach to the multiple-depot vehicle scheduling problem. *Operations Research*, 42(1):41–52, 1994. doi: 10.1287/opre.42.1.41. URL <https://doi.org/10.1287/opre.42.1.41>.

Ons Sassi and Ammar Oulamara. Electric vehicle scheduling and optimal charging problem: Complexity, exact and heuristic approaches. *International Journal of Production Research*, 55, 03 2014. doi: 10.1080/00207543.2016.1192695.

El-Ghazali Talbi. *Metaheuristics: From Design to Implementation*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2009. ISBN 978-0-470-27858-1. Includes bibliographical references and index.

Chunlu Wang, Congcong Guo, and Xingquan Zuo. Solving multi-depot electric vehicle scheduling problem by column generation and genetic algorithm. *Applied Soft Computing*, 112:107774, 2021a. ISSN 1568-4946. doi: <https://doi.org/10.1016/j.asoc.2021.107774>. URL <https://www.sciencedirect.com/science/article/pii/S1568494621006955>.

Lixing Wang, Zhenning Wu, and Changyong Cao. Integrated optimization of routing and energy management for electric vehicles in delivery scheduling. *Energies*, 14, 2021b. doi: 10.3390/en14061762.

Guang-Jing Zhou, Dong-Fan Xie, Xiao-Mei Zhao, Chaoru Lu, Cheng-Dong Zhou, and Chao-Yang Zhang. Collaborative optimization of vehicle scheduling and speed regulation for autonomous customized bus systems. *Applied Mathematical Modelling*, 128:410–430, 2024. ISSN 0307-904X. doi: <https://doi.org/10.1016/j.apm.2024.01.032>. URL <https://www.sciencedirect.com/science/article/pii/S0307904X24000325>.