

Um Algoritmo Genético de Chaves Aleatórias para resolver Problema de Roteamento de Veículos Elétricos e de Localização de Estações de Troca de Baterias com Múltiplos Depósitos e Janela de Tempo

**Bráulio Manoel de Oliveira Portela, Carlos Cristiano Hasen clever Borges,
Heder Soares Bernardino, Luciana Brugiolo Gonçalves, Stênio Sã Rosário F. Soares**

Universidade Federal de Juiz de Fora

Rua José Lourenço Kelmer - Campus Universitário, Juiz de Fora - MG

{braulio.portela, cchborges, heder,
lbrugiolo, ssoares}@ice.ufjf.br

RESUMO

O uso de veículos elétricos em logística de transporte tem se tornado uma realidade. Este novo cenário demanda algoritmos que consideram autonomia e tempo de abastecimento do veículo. Neste trabalho é proposto um Algoritmo Genético de Chaves Aleatórias para o Problema de Roteamento de Veículos Elétricos e de Localização de Estações de Troca de Baterias com Múltiplos Depósitos e Janela de Tempo, onde objetiva-se minimizar a distância total percorrida pela frota para atendimento aos clientes quanto à demanda e janela de tempo, devendo decidir quais depósitos serão utilizados e em quais locais serão instaladas estações de troca de bateria. O algoritmo proposto faz uso de uma fase de refinamento que utiliza *Variable Neighborhood Descent* com seis padrões de busca local e foi submetido a um conjunto de instâncias e os resultados comparados com os obtidos por uma abordagem da literatura. Os experimentos demonstram a eficácia do algoritmo proposto.

PALAVRAS CHAVE. Algoritmo Genético de Chaves Aleatórias. Veículos elétricos com troca de bateria. Variable Neighborhood Descent.

L&T – Logística e Transportes, MH – Metaheurísticas, OC – Otimização Combinatória

ABSTRACT

The use of electric vehicles in transport logistics has become a reality. This new scenario demands algorithms that consider the vehicle's autonomy and fueling time. In this work, a Random-Key Genetic Algorithm is proposed for the Multi-Depot Electric Vehicle Location Routing Problem with Time Windows and Battery Swapping, where the objective is to minimize the total distance traveled by the fleet to serve customers in terms of demand and time window. It must also decide which depots will be used and in which places battery change stations will be installed. The proposed algorithm makes use of a refinement phase that uses Variable Neighborhood Descent with six local search patterns and was submitted to a set of instances and the results compared with those obtained by a literature approach. The experiments demonstrate the effectiveness of the proposed algorithm.

KEYWORDS. Random-Key Genetic Algorithms. Electric vehicles with battery swapping. Variable Neighborhood Descent.

L&T – Logistic and Transport, MH – Metaheuristics, CO – Combinatorial Optimization

1. Introdução

A redução dos impactos ambientais tem se tornado uma preocupação presente no dia a dia de muitas empresas, agências e pesquisadores. Esse pensamento levou ao surgimento de uma área de pesquisa denominada *Green Logistics* [Sbihi e Eglese, 2010], que visa buscar e viabilizar alternativas com menores danos ao meio ambiente. A utilização de veículos elétricos (VE) em problemas de roteamento traz novos desafios nesta área.

Recentemente, um cenário favorável ao uso dos veículos elétricos tem sido gradativamente criado devido aos incentivos governamentais [Langbroek et al., 2016], principalmente em países europeus e asiáticos, e ao desenvolvimento de novas tecnologias de baterias, com maior capacidade de armazenamento de energia, maior velocidade de carregamento e preços competitivos [Castro e Ferreira, 2010]. No entanto, a autonomia dos veículos elétricos permanece relativamente menor quando comparada com veículos com motores a combustão interna, o que consiste em um desafio maior para o planejamento de rotas [Mirchandani et al., 2014].

Na cadeia de suprimentos, a menor autonomia dos veículos elétricos e o tempo necessário para recarga de baterias ainda dificultam a substituição dos veículos de combustão interna. Além desses aspectos, um outro fator importante a considerar é a necessidade de instalação facilidade, como postos de recarga ou de troca de baterias.

A instalação de estações de troca de bateria (BSS), do inglês *Battery Swapping*, em pontos estratégicos é uma alternativa para possibilitar que o veículo percorra longas distâncias, conforme mostrado por Mirchandani et al. [2014]. Em vez de gastar um tempo considerável para recarregar a bateria ao longo do trajeto, a troca da bateria por outra gastaria poucos minutos, causando um menor impacto no tempo total da viagem daquele veículo. Em aplicações onde se considera janela de tempo, esta questão é determinante no que se refere ao espaço de viabilidade do problema.

Paz et al. [2018] apresentam o Problema de Roteamento e de Localização de Veículos Elétricos com Múltiplos Depósitos e Janela de Tempo (MDEVLRPTW), cujo foco é determinar a sequência de atendimento a um conjunto de clientes com demanda e janela de tempo por meio de uma frota de veículos elétricos, que podem partir de um conjunto de depósitos. Além disso, o problema requer a definição de locais para a instalação da infraestrutura necessária ao reabastecimento de energia dos veículos. O objetivo é otimizar uma função de custo, neste caso, distância total percorrida. Os autores propõem três modelos de Programação Linear Inteira, que diferem entre si quanto ao tipo de tecnologia de reabastecimento de veículo (recarga, troca de bateria, ou mista) a ser utilizada como facilidade e mostram resultados para instâncias com 5, 10 e 15 clientes.

Portela et al. [2021] abordam o MDEVLRPTW-BS, modelo em que as facilidades são estações de troca de bateria e apresentam resultados para o conjunto de instâncias de Paz et al. [2018], além de um outro conjunto de instâncias com até 360 clientes.

O objetivo deste trabalho é propor uma nova abordagem para o MDEVLRPTW-BS com estações de troca de bateria e analisar sua eficácia em relação às duas abordagens da literatura. A proposta consiste num RKGA-VND, uma técnica que usa um Algoritmo Genético de Chaves Aleatórias (RKGA) [Gonçalves e Resende, 2011] com decodificador que se utiliza do Algoritmo de Inserção Mais Barato e uma Busca Gulosa. Por fim, um conjunto elite de soluções obtidas é submetido ao procedimento *Variable Neighborhood Descent* (VND) [Hansen et al., 2010], para refinamento.

O restante deste trabalho está assim definido, então assim apresentado: na Seção 2 é apresentado uma visão geral de problemas relacionados e trabalhos da literatura; a Seção 3 trata da abordagem proposta e os métodos utilizados; na Seção 4 são mostrados os experimentos computacionais, os parâmetros utilizados, os resultados obtidos e uma breve análise dos mesmos; e por fim,

as conclusões e trabalhos futuros são apresentados na Seção 5.

2. Trabalhos relacionados

Normalmente, dois problemas de otimização são estudados quando se utilizam veículos elétricos: (i) o problema de roteamento do veículo [Toth e Vigo, 2014], que é apresentado pela primeira vez por Dantzig e Ramser [1959] e (ii) a designação dos locais das instalações de facilitadores [Mara et al., 2021], sendo estes pontos de recarga ou de troca de bateria. Neste estudo, o Problema de Roteamento de Veículos Elétricos e de Localização de Estações de Troca de Baterias com Múltiplos Depósitos e Janela de Tempo (MDEVLRPTW-BS) é tratado, que é a combinação dos dois tipos de problema citados. Nesta seção, é apresentada uma breve revisão dos problemas clássicos e de localização e roteamento.

2.1. Problema de roteamento e de localização

O problema de localização, quando combinado com o problema de roteiro, consiste em determinar, entre as localizações candidatas, aquelas que melhor servem como pontos de apoio. Esses locais podem ser qualquer tipo de instalação, como depósitos de mercadorias, uma sede de empresa e um hotel [Mara et al., 2021]. Neste trabalho, um MDEVLRPTW é atacado e, portanto, a instalação é uma estação.

O problema de roteamento e localização (LRP) combina esses dois eixos de decisão, que devem ser resolvidos simultaneamente. Nas últimas décadas, muitas variantes do LRP foram propostas, tais como: LRP com vários depósitos [Laporte et al., 1988; Wu et al., 2002], LRP capacitado (com capacidade de depósitos ou veículos limitada) [Lysgaard et al., 2004; Lin et al., 2009] e LRP com janela de tempo [Zarandi et al., 2011, 2013].

Existem alguns estudos voltados para a designação de pontos de reabastecimento de veículos de combustão interna, assunto que vem ganhando muita atenção recentemente. Com base no *Flow Capture Location Model* (FCLM) proposto por Hodgson [1990]; Berman et al. [1992], Kuby e Lim estudaram o Flow-Refueling Location Model (FRLM) no qual as estações estão localizadas para maximizar o fluxo de veículos nos caminhos [Kuby e Lim, 2007].

O problema de roteamento e localização de estações de troca de bateria de veículo elétrico (BSS-EV-LRP) foi proposto por Yang e Sun [2015] e visa determinar a localização de estações de troca de bateria e o plano de roteamento de uma frota de veículos elétricos. As BSS são determinadas estrategicamente, considerando um custo de implementação, dado um conjunto de pontos candidatos. O objetivo do problema é minimizar a soma do custo de implantação do BSS com os custos de energia dos veículos na realização de suas rotas.

Yang e Sun [2015] apresentam modelos para o problema e duas combinações de técnicas de otimização. Além disso, uma profunda análise econômica e ambiental também é realizada, incluindo a comparação entre os cenários básico e ampliado, mostrando a eficiência e redução de emissões na utilização de VEs nas práticas logísticas.

Outra combinação de ambos os tipos de problemas de otimização é proposta por Li-ying e Yuan-bin [2015]. Os autores consideram que cada estação pode ter diferentes tipos de tecnologia. O tipo de estação depende do tempo de carga da bateria (uma função linear que depende do tipo de estação e da quantidade de energia restante), do custo de construção e do custo da eletricidade. Além disso, consideraram a recarga nos vértices do cliente, posicionando os pontos candidatos para localizar estações de recarga com as mesmas coordenadas do cliente e também consideraram restrições de janela de tempo. Uma combinação heurística que incorpora um *Adaptive Variable Neighborhood Search* (AVNS) com o algoritmo *Tabu Search* para refinamento foi usado para resolver o problema.

Paz et al. [2018] apresentam uma extensão dos modelos de roteamento de localização usando veículo elétrico para cenários com múltiplos depósitos. Aspectos importantes já conhecidos

na literatura também foram considerados, como restrições de janela de tempo, a possibilidade de recarga parcial, a possibilidade de múltiplas visitas aos vértices, a possibilidade de localização de estações em todos os vértices (incluindo depósitos), e a combinação dos recarga de bateria convencional e BSS. Para validação dos modelos propostos, um conjunto de instâncias baseadas nas de Schiffer e Walther [2017] foi proposto e usado no resolvedor Cplex.

Portela et al. [2021] trabalharam este mesmo problema e apresentaram duas técnicas utilizando GRASP e duas heurísticas construtivas diferentes, que apresentaram resultados promissores e tempos de processamento baixos quando comparados a outros trabalhos da literatura. Além disso foram apresentadas instâncias de alta dimensionalidade que também serão utilizadas neste trabalho para fins comparativos.

Uma outra abordagem heurística, focada em um Algoritmo Genético de Chaves Aleatórias, é proposta aqui para resolver o primeiro modelo de Paz et al. [2018], que consiste em usar estações de troca de bateria (BSS) como uma facilidade para VEs.

2.2. Problema MDEVLRPTW-BS

O Problema de Roteamento de Veículos Elétricos e de Localização de Estações de Troca de Baterias com Múltiplos Depósitos e Janela de Tempo (MDEVLRPTW-BS) [Paz et al., 2018] usa estações de troca de bateria (BSS) como facilitadores no problema de roteamento. Este problema considera (i) um conjunto de depósitos candidatos, (ii) um conjunto de clientes com demandas a serem atendidas, janelas de tempo para recebimento de entregas e um tempo necessário para que as entregas sejam executadas, (iii) a frota homogênea de veículos elétricos com capacidade de carga e bateria, (iv) um conjunto de locais candidatos para instalação de BSS. Os locais de início e fim de qualquer rota são um dos depósitos selecionados.

O objetivo do MDEVLRPTW-BS é minimizar a distância total percorrida pelos veículos para atender todos os clientes sem comprometer as janelas de tempo e as capacidades de carga e bateria dos veículos obedecendo as seguintes restrições: (i) cada cliente deve ser visitado por apenas um veículo, (ii) cada veículo deve retornar ao seu depósito inicial, (iii) uma BSS não tem limite de veículos atendidos em paralelo e a bateria nova está sempre cheia, (iv) o intervalo de tempo necessário para trocar a bateria é fixo, e (v) o tempo de viagem entre dois locais é calculado pela distância entre eles dividido pela velocidade média do veículo.

3. Abordagem proposta

Para resolver o Problema de Roteamento de Veículos Elétricos e de Localização de Estações de Troca de Baterias com Múltiplos Depósitos e Janela de Tempo (MDEVLRPTW-BS), o método RKGA-VND proposto considera uma abordagem com Algoritmo Genético de Chaves Aleatórias (RKGA) [Gonçalves e Resende, 2011] combinado com *Variable Neighborhood Descent* (VND) como estratégia de pós-otimização. Como decodificador, foi utilizado o algoritmo de Inserção mais barata para definir a ordem de visitação dos clientes e uma busca gulosa para inserção das estações de troca de baterias.

3.1. RKGA-VND

O Algoritmo Genético de Chaves Aleatórias (RKGA), introduzido por Bean [1994], é um Algoritmo Genético onde a representação dos indivíduos é feita de forma indireta, ou seja, é necessário um decodificador para determinar o fenótipo das soluções. Vetores com chaves reais, com valores definidos no intervalo contínuo [0,1), são usados para representar um indivíduo. Na população inicial, estas chaves que compõem os indivíduos são geradas de forma aleatória. A Figura 1 ilustra esse tipo de representação, onde há uma chave para cada cliente a ser visitado na solução (o tamanho do vetor é igual ao número de clientes da instância). Neste trabalho, uma

0	1	2	3	4	5	6	7	8	9
0.38	0.11	0.49	0.91	0.51	0.38	0.01	0.54	0.60	0.45

Figura 1: Ilustração de um vetor de Chaves Aleatórias gerado para uma instância com 10 clientes.

combinação do algoritmo de Inserção Mais Barata e uma busca gulosa para a inserção de estações BSS é usada como decodificador (descrito na Seção 3.2).

Para calcular a aptidão dos indivíduos, estes são submetidos ao decodificador. O decodificador identifica a solução representada no indivíduo e determina seu custo. Em seguida, os indivíduos da população são ordenados segundo a aptidão (custo).

Na sequência, indivíduos são selecionados para recombinação e mutação. O processo de seleção é realizado via torneio binário, onde dois indivíduos são sorteados aleatoriamente da população, e é escolhido entre eles o com menor valor de aptidão. Como operador de recombinação adotou-se o cruzamento de um único ponto [Dréo et al., 2006]. Para mutação, realiza-se a troca entre dois elementos do cromossomo escolhidos aleatoriamente.

Para gerar a população para a próxima geração, mantém-se a elite da população corrente e o restante da população vem dos processos de recombinação e mutação. Este processo é finalizado quando se atinge o número pré-estabelecido de gerações.

Após o processo evolutivo, é feito o refinamento final com o VND (descrito na Seção 3.3). O VND é aplicado a um subconjunto da população final, que neste caso é formado por 20% dos melhores indivíduos da população. Após o refinamento ser executado em todos os indivíduos do subconjunto, a melhor solução obtida será a solução final do método proposto.

3.2. Decodificador

Dada uma solução representada de forma indireta via chaves aleatórias, como ilustrado na Figura 1, o objetivo do decodificador é determinar a solução associada ao MDEVRPTW-BS. Para representar a solução decodificada do problema, foi utilizado um vetor de rotas onde cada rota é uma lista de adjacências. A Figura 2 apresenta um exemplo desta estrutura, onde D_i , S_j e C_k denotam o i -ésimo depósito, j -ésimo BSS e k -ésimo cliente, respectivamente.

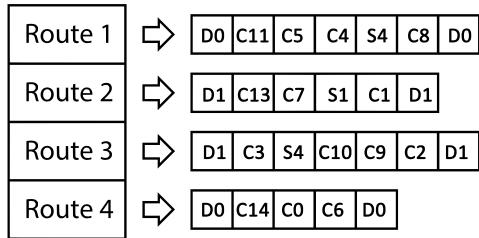


Figura 2: Ilustração de um vetor de vetores para uma solução de multi-roteamento por listas de adjacência.

O mecanismo de decodificação utilizado é dividido em duas etapas. Primeiro, as rotas são construídas em paralelo, usando o método de Inserção mais barata. Em seguida, as BSS são localizadas de acordo com a necessidade de cada rota. Um pseudo-código deste método é apresentado Algoritmo 1.

O processo de construção de rotas começa na linha 2, onde se estima o número mínimo de veículos necessário para atender os clientes. Para isso, soma-se as demandas q_i de todos os clientes e divide-se pela capacidade U dos veículos. O teto deste valor será associado ao número de veículos necessários, ve . Em seguida, na linha 3, uma solução é criada com número de rotas igual a ve .

Na linha 4 é criada a lista de candidatos LC , composta por todos os clientes. Os clientes são ordenados nesta lista de forma decrescente segundo o valor associado no vetor de chaves aleatórias. Em seguida, linha 5, os primeiros clientes da lista são acrescentados como clientes iniciais em cada rota e removidos de LC . A partir do momento em que cada rota recebeu um cliente, que chamaremos de pivô, cada rota recebe então os vértices inicial e final: o depósito. Para cada rota, o depósito mais próximo do cliente pivô é adicionado.

A iteração na Linha 6 se repete enquanto a lista de clientes LC não estiver vazia. Na linha 7, i' recebe o próximo cliente com maior Chave Aleatória ainda pertencente à lista LC . Em seguida, linha 8, i' é inserido em S na melhor rota e na melhor posição (menor impacto na aptidão da solução), sem violar as janelas de tempo e as restrições de capacidade do veículo. Na linha 9 este cliente é removido do conjunto LC .

Algorithm 1: Decodificador.

```

1 begin
2   |    $ve \leftarrow \lceil \sum_{i \in I} q_i / U \rceil$ ;
3   |    $S \leftarrow initSol(ve)$ ;
4   |    $LC \leftarrow I$ ;
5   |    $S \leftarrow addInitialCustomer(S, LC)$ ;
6   |   while  $LC.size() > 0$  do
7   |     |    $i' \leftarrow nextCustomer(LC)$ ;
8   |     |    $S \leftarrow bestInsertion(S, i')$ ;
9   |     |    $LC \leftarrow LC - \{i'\}$ ;
10  |    $IBSS(S)$ ;
11  |   return  $S$ ;

```

Dentro da função $nextCustomer(S, i')$, uma nova rota é criada em S quando i' viola qualquer restrição de capacidade do veículo ou janelas de tempo em quaisquer das rotas existentes. Nesse caso, i' é alocado nesta nova rota e o depósito usado é o mais próximo.

Quando o conjunto LC estiver vazio, a iteração é finalizada e é executado o processo de inserção de BSS (linha 10). Para selecionar as BSS utilizadas na solução, são investigados os pontos de ruptura das rotas. Esses pontos ocorrem quando um veículo esgotou sua bateria e não consegue chegar ao próximo cliente. Todas as estações candidatas alcançadas a partir do último cliente visitado antes da bateria descarregar, verificando o restante de bateria e as disponibilidades das janelas de tempo, são adicionadas a um conjunto e deste é escolhida a mais próxima para ser adicionada a rota.

Sempre que não existir nenhuma BSS capaz de atender ao veículo atual é realizado um retrocesso na rota. Assim, volta-se um cliente na rota e outra pesquisa é feita para verificar as disponibilidades de estações de recarga adequadas e é feita a escolha da mais próxima, quando houver. Observe que este procedimento pode iterar sem que uma estação de recarga seja encontrada, retrocedendo até o depósito sem encontrar nenhuma estação candidata viável. Nesse caso, a solução atual é descartada e, em seguida, uma nova solução é inicializada com uma rota adicional no passo de estimativa de quantidade de rotas ao inicio do construtivo. Como resultado do Algoritmo 1 é retornada a solução e seu respectivo valor de aptidão.

3.3. VND

O *Variable Neighborhood Descent* consiste numa metaheurística que explora de forma sistemática um conjunto de estruturas de vizinhança. Para compor o VND, as mesmas seis vizinhanças de Portela et al. [2021] foram utilizadas. Essas vizinhanças e a ordem de execução no VND são mostradas na Tabela 1. Enquanto as buscas *Inter-routes* operam entre mais de uma rota, as *Intra-routes* operam em cada rota individualmente. Testes preliminares foram realizados para determinar a melhor ordem para as vizinhanças, onde observou-se a maior contribuição das mesmas.

Tabela 1: [Lista da ordem das buscas locais utilizadas no VND.](#)

#	Local Searches	Range
1	<i>Shift(1,0)</i>	Inter-routes
2	<i>Reinsertion</i>	Intra-routes
3	<i>Isolate</i>	Intrer-routes
4	<i>2-opt</i>	Intra-routes
5	<i>ChangeBSS</i>	Intra-routes
6	<i>Swap(1,1)</i>	Inter-routes

O movimento *Shift(1,0)* [Osman, 1993] consiste em mudar um cliente de uma rota para outra. A busca *Reinsertion* [Or, 1976] reinsere um cliente em outro lugar na mesma rota. Já o movimento *Isolate* simplesmente remove o cliente de sua rota e cria uma nova rota apenas para ele, com a saída e retorno sendo o depósito mais adequado. No método *2-opt* [Croes, 1958], duas arestas não adjacentes são substituídas por duas novas arestas formando uma nova rota. O movimento *ChangeBSS* tenta substituir uma estação estabelecida em uma rota por outra, sendo adicionada na melhor posição possível e mantendo a viabilidade da rota. E, por fim, a busca *Swap(1,1)* [Osman, 1993] muda a posição de dois clientes de rotas diferentes. Para mais detalhes sobre as vizinhanças utilizadas, ver em Portela et al. [2021].

4. Experimentos Computacionais

A abordagem proposta foi implementada utilizando a linguagem de programação C++ e o código fonte está disponível publicamente¹. Os experimentos foram realizados em uma máquina virtual com CPU Intel (R) Xeon (R) E7530 @ 1,87 GHz, 16 GB de RAM, no Ubuntu 18.04.4.

Como critério de parada, adotou-se 10 gerações sem atualização da melhor solução. O tamanho da população foi ajustado em $5n$, sendo n a quantidade de clientes de cada instância, limitado a 100 indivíduos. Foram realizadas 10 execuções independentes para cada instância. O cruzamento de um único ponto tem chance de 80% de ocorrer, e caso não ocorra, são mantidos os dois indivíduos sorteados para a próxima população. Os indivíduos gerados pelo cruzamento, por sua vez, têm 95% de chance de sofrerem mutação e dois valores aleatórios de dois genes são trocados entre si. Por fim, na nova geração, 10% dos piores indivíduos são descartados e substituídos pelos melhores 10% da geração anterior.

Entre as instâncias consideradas, as primeiras 36, de menor porte, são modificações das quais usadas por Paz et al. [2018] para o problema de roteamento com estações de carregamento. As instâncias estão disponíveis na web² e possuem 5, 10 e 15 clientes. As outras 36 instâncias, de maior porte, com 100, 144, 288 e 360 clientes são apresentadas por Portela et al. [2021]. Essas instâncias também estão disponíveis¹.

¹<https://github.com/brmportela/MDEVLRPTW/>

²<http://academia.utp.edu.co/planeamiento/?p=3561>

4.1. Análise comparativa dos resultados

Os resultados obtidos pelo RKGA-VND, quando aplicado à instâncias com 5, 10 e 15 clientes, são mostrados na Tabela 2. Para estas instâncias há na literatura, por Paz et al. [2018], o uso de resolvedor CPLEX, limitado a 28800 segundos. Este resultado está referenciado na Tabela 2 nas colunas rotuladas por CPLEX. Além deste resultado, há um comparativo com os resultados encontrados pela técnica GRASP-CI-RVND de Portela et al. [2021]. Para cada abordagem e cada instância, a Tabela 2 mostra o custo (distância percorrida) da melhor solução (d^{best}) e média dos custos para as 10 execuções (d^{avg}), bem como a média (t^{avg}) dos tempos de CPU (em segundos) demandado em cada abordagem. A coluna Δd apresenta a diferença percentual relativa entre os resultados obtidos pela proposta RKGA-VND e o melhor resultado da literatura. Os melhores resultados são destacados em negrito. Além disso, são apresentadas as seguintes características das soluções: o número de depósitos usados (dp), o número de veículos ou rotas designados (vh), e o número de estações visitadas (st).

Tabela 2: Resultados experimentais do algoritmo RKGA-VND em comparação com CPLEX [Paz et al., 2018] e GRASP-CI-RVND [Portela et al., 2021] para 36 instâncias de 5, 10 e 15 clientes.

INPUTS		CPLEX		GRASP-CI-RVND				RKGA-VND						
n	Instance	dp/vh/st	dist.	time	dp/vh/st	d^{best}	d^{avg}	t^{avg}	dp/vh/st	d^{best}	d^{avg}	t^{avg}	$\Delta d(\%)$	
5	c101-5	2/4/1	238.62	0.30	2/4/1	238.62	238.61	0.00	2/4/1	238.62	238.61	0.01	0.00	
	c103-5	1/3/0	165.67	0.55	1/2/1	165.67	165.67	0.01	1/2/1	165.67	165.67	0.01	0.00	
	c206-5	2/4/1	234.31	2.14	2/3/2	234.31	234.31	0.00	2/4/1	234.31	234.31	0.05	0.00	
	c208-5	1/2/1	129.13	1.09	1/2/1	129.13	129.13	0.00	1/2/1	129.13	129.13	0.01	0.00	
	r104-5	2/2/1	118.51	0.61	2/2/1	118.51	118.51	0.00	2/2/1	118.51	118.51	0.00	0.00	
	r105-5	2/2/1	104.70	0.18	2/2/1	104.70	104.70	0.00	2/2/1	104.70	104.70	0.00	0.00	
	r202-5	1/1/3	126.90	1.85	1/1/2	126.90	126.90	0.00	1/1/2	126.90	126.90	0.00	0.00	
	r203-5	2/2/2	167.33	0.63	2/2/2	167.33	167.33	0.00	2/2/2	167.33	167.33	0.01	0.00	
	rc105-5	2/3/1	185.72	0.66	2/3/1	185.72	185.72	0.00	2/3/1	185.72	185.72	0.01	0.00	
	rc108-5	2/2/2	242.10	7.09	2/2/2	242.09	242.09	0.00	2/2/2	242.09	242.09	0.01	0.00	
	rc204-5	1/1/3	176.39	38.15	1/1/3	176.39	176.39	0.00	1/1/3	176.39	177.27	0.00	0.00	
	rc208-5	1/1/2	167.98	9.57	1/1/2	167.98	167.98	0.00	1/1/2	167.98	167.98	0.00	0.00	
	Avg.		171.45	5.24		171.45	171.45	0.00		171.45	171.52	0.01	0.00	
10	c101-10	2/4/3	331.10	582.72	2/4/2	335.22	335.22	0.30	2/5/2	337.45	340.19	0.03	1.92	
	c104-10	2/2/3	243.37	41.99	2/2/3	243.37	243.37	0.02	2/2/3	243.37	265.77	0.04	0.00	
	c202-10	2/3/2	227.25	27.45	2/3/2	227.25	227.45	0.09	2/2/2	230.16	230.16	0.04	1.28	
	c205-10	2/3/1	204.44	1.64	2/3/1	204.44	204.44	0.00	2/3/1	204.44	204.44	0.03	0.00	
	r102-10	2/3/3	232.37	28.58	2/3/3	232.37	232.37	0.02	2/3/3	232.37	232.37	0.03	0.00	
	r103-10	2/3/2	193.41	9747.56	2/3/2	193.41	193.41	0.03	2/3/2	193.41	194.04	0.04	0.00	
	r201-10	2/4/2	206.74	16.06	2/4/2	206.74	206.74	0.05	2/4/2	206.74	207.69	0.04	0.00	
	r203-10	1/1/4	216.99	10880.50	1/1/4	216.99	216.99	0.02	1/1/4	216.99	226.11	0.05	0.00	
	rc102-10	2/5/1	386.96	18.63	2/5/1	386.96	386.96	0.01	2/5/1	386.96	386.96	0.04	0.00	
	rc108-10	2/4/2	312.04	89.41	2/4/2	312.04	312.04	0.00	2/4/2	312.04	312.04	0.05	0.00	
	rc201-10	2/4/1	298.00	94.39	2/4/1	298.00	298.00	0.00	2/4/1	298.00	298.00	0.04	0.00	
	rc205-10	2/3/2	290.87	3.93	2/3/2	290.87	290.87	0.00	2/3/2	290.87	290.87	0.03	0.00	
	Avg.		261.96	1794.41		262.31	262.32	0.05		262.73	265.72	0.04	0.27	
15	c103-15	2/3/2	320.17	28800.00	2/3/2	320.17	320.17	0.90	2/3/2	320.17	320.17	0.17	0.00	
	c106-15	2/4/1	261.20	68.45	2/4/1	261.20	261.20	0.03	2/4/1	261.20	266.77	0.13	0.00	
	c202-15	2/3/4	353.75	1104.17	2/3/4	353.75	353.75	0.21	2/3/4	353.75	357.67	0.24	0.00	
	c208-15	1/2/3	300.55	635.45	1/2/3	300.55	302.66	0.47	1/2/3	300.55	301.64	0.22	0.00	
	r102-15	2/5/3	335.47	2293.45	3/5/2	335.47	337.23	0.48	2/5/3	335.47	337.00	0.22	0.00	
	r105-15	2/4/3	289.00	17.43	2/4/3	289.00	292.67	0.37	2/4/3	289.00	289.00	0.20	0.00	
	r202-15	2/3/4	323.18	5422.84	2/3/4	323.18	323.18	0.09	2/3/4	323.18	327.64	0.20	0.00	
	r209-15	1/2/4	274.34	1258.39	1/2/4	274.34	278.98	281.19	0.99	2/2/4	280.31	285.48	0.20	2.18
	rc103-15	2/5/3	354.26	28800.00	2/5/3	354.26	354.26	0.86	2/5/3	354.26	355.39	0.18	0.00	
	rc108-15	1/3/4	372.54	28800.00	1/3/4	370.25	373.15	0.96	1/3/3	373.24	380.46	0.28	0.19	
	rc202-15	2/3/3	370.76	800.80	2/3/3	370.76	370.76	0.23	2/4/3	370.76	373.02	0.30	0.00	
	rc204-15	1/2/4	310.58	28800.00	1/2/4	310.58	310.58	0.01	1/2/4	310.57	310.57	0.21	0.00	
	Avg.		322.15	10566.75		322.35	323.40	0.47		322.71	325.40	0.21	0.20	

Pode-se observar na Tabela 2 que tanto a abordagem GRASP-CI-RVND quanto o RKGA-VND proposto obtiveram resultados muito próximos aos apresentados pelo resolvedor Cplex. Em

apenas quatro instâncias o RKGA-VND não alcançou a melhor solução, com variações de no máximo 2.18%. Nestes quatro casos, observa-se uma diferença nas soluções do AG em termos de número de veículos, número de estações ou de depósitos.

É importante destacar que a abordagem proposta obteve estes resultados com um tempo médio de CPU significativamente menor quando comparado ao CPLEX e bem próximos dos observados pelo GRASP-CI-RVND. Além disso, nos casos em que a proposta não encontrou os melhores resultados, os tempos de CPU necessários para a resolução dos problemas são menores que os do CPLEX. Em muitos dos casos, onde os tempos foram de 0.00s, estes representam que a abordagem chegou ao resultado com tempo menor do que 0.005 segundos.

Para uma melhor avaliação da abordagem e para demonstrar a eficiência de um método heurístico em relação às instâncias onde um modelo exato com um resolvedor não consegue obter resultados em tempo viável, um novo conjunto de instâncias de tamanhos maiores foi apresentado por Portela et al. [2021]. Foram utilizados para os experimentos os conjuntos de instâncias que possuem 100, 144, 288 e 360 clientes com 4 ou 6 depósitos. Os resultados obtidos para essas instâncias são apresentados na Tabela 3, que apresenta os mesmos elementos da Tabela 2, só que apenas para as duas abordagens heurísticas. Pode-se observar que, para 28 das 36 instâncias maiores (100, 144, 288 e 360), a abordagem RKGA-VND apresentou melhores resultados do que a abordagem GRASP-CI-RVND, indicando uma superioridade da abordagem proposta em relação à da literatura, baseada na metaheurística GRASP.

Vale destacar também a coluna Δd , que mostra a diferença percentual entre as soluções das duas abordagens, onde, principalmente para as instâncias de 360 clientes, RKGA-VND se mostrou consideravelmente melhor, com -14,03% na média final.

Outro aspecto interessante a ser destacado que se observou é que para os conjuntos de instâncias maiores (Tabela 3) o RKGA-VND demonstra uma capacidade de solucionar o problema com menos veículos do que o GRASP-CI-RVND. Esta diferença é cada vez maior em relação ao tamanho das instâncias utilizadas, chegando até uma diferença de 53 veículos utilizados pelo RKGA-VND para 87 utilizados pelo GRASP-CI-RVND para a instância pr11a-c de 360 clientes, uma variação percentual de mais de 60%.

5. Conclusões e Trabalhos Futuros

Neste artigo, um Algoritmo Genético de Chaves Aleatórias com VND (RKGA-VND) é proposto para o Problema de Roteamento e Localização de Veículos Elétricos com Múltiplos Depósitos e Janela de Tempo e Troca de Bateria (MDEVRPTW-BS). Para a decodificação das chaves, o algoritmo de inserção mais barata foi usado para montar as rotas juntamente com uma busca para localizar as estações. Um VND com seis buscas locais foi utilizado para refinamento sobre a elite da população final. O problema resolvido aqui encontra aplicação no contexto de viabilizar economicamente o uso de veículos elétricos na logística, uma das áreas de pesquisa da Logística Verde.

Experimentos computacionais foram realizados usando instâncias da literatura. Os resultados obtidos para instâncias menores (5, 10 e 15) mostraram que o RKGA-VND é uma alternativa viável quando comparada aos resultados da literatura (CPLEX e GRASP-CI-RVND) alcançando o resultado ótimo em quase todos os casos.

A abordagem proposta se mostrou competitiva também para instâncias maiores quando comparada à abordagem GRASP-CI-RVND da literatura. E uma vez que tanto a abordagem proposta quanto à abordagem GRASP da literatura fazem uso da heurística de inserção mais barata para construir as rotas e usam estruturas de vizinhança semelhantes, presume-se que o uso de um

procedimento evolutivo com chaves randômicas pode ser um caminho promissor para problemas desta natureza.

Para trabalhos futuros, um bom caminho é a criação de outros decodificadores para o AG de modo a melhorar o desempenho do método proposto aqui. Além disso, pretende-se atacar instâncias maiores, por exemplo, com 420 e 600 clientes. Por fim, um modelo alternativo considerando objetivos múltiplos ou uma estrutura de dois níveis também é uma direção de pesquisa interessante a ser investigada.

Agradecimentos

Os autores agradecem o apoio financeiro do CNPq, Capes, FAPEMIG e UFJF.

Referências

- Bean, J. C. (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA journal on computing*, 6(2):154–160.
- Berman, O., Larson, R. C., e Fouska, N. (1992). Optimal location of discretionary service facilities. *Transportation Science*, 26(3):201–211.
- Castro, B. H. R. d. e Ferreira, T. T. (2010). Veículos elétricos: aspectos básicos, perspectivas e oportunidades. *BNDES Setorial*, n. 32, set. 2010, p. 267-310.
- Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations research*, 6 (6):791–812.
- Dantzig, G. B. e Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1): 80–91.
- Dréo, J., Pétrowski, A., Siarry, P., e Taillard, E. (2006). *Metaheuristics for hard optimization: methods and case studies*. Springer Science & Business Media.
- Gonçalves, J. F. e Resende, M. G. (2011). Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525.
- Hansen, P., Mladenović, N., e Pérez, J. M. (2010). Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175(1):367–407.
- Hodgson, M. J. (1990). A flow-capturing location-allocation model. *Geographical Analysis*, 22(3): 270–279.
- Kuby, M. e Lim, S. (2007). Location of alternative-fuel stations using the flow-refueling location model and dispersion of candidate sites on arcs. *Networks and Spatial Economics*, 7(2):129–152.
- Langbroek, J. H., Franklin, J. P., e Susilo, Y. O. (2016). The effect of policy incentives on electric vehicle adoption. *Energy Policy*, 94:94–103.
- Laporte, G., Nobert, Y., e Taillefer, S. (1988). Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation science*, 22(3):161–172.
- Li-ying, W. e Yuan-bin, S. (2015). Multiple charging station location-routing problem with time window of electric vehicle. *Journal of Engineering Science & Technology Review*, 8(5).

- Lin, S.-W., Lee, Z.-J., Ying, K.-C., e Lee, C.-Y. (2009). Applying hybrid meta-heuristics for capacitated vehicle routing problem. *Expert Systems with Applications*, 36(2):1505–1512.
- Lysgaard, J., Letchford, A. N., e Eglese, R. W. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2):423–445.
- Mara, S. T. W., Kuo, R., e Asih, A. M. S. (2021). Location-routing problem: a classification of recent research. *International Transactions in Operational Research*, 28(6):2941–2983. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.12950>.
- Mirchandani, P., Adler, J., e Madsen, O. B. (2014). New logistical issues in using electric vehicle fleets with battery exchange infrastructure. *Procedia-Social and Behavioral Sciences*, 108:3–14.
- Or, I. (1976). Traveling salesman problem-type combinatorial optimization problems and their relation to the logistics of regional blood banking. *Northwestern University, Evanston, IL*.
- Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of operations research*, 41(4):421–451.
- Paz, J., Granada-Echeverri, M., e Escobar, J. (2018). The multi-depot electric vehicle location routing problem with time windows. *International Journal of Industrial Engineering Computations*, 9(1):123–136.
- Portela, B. M., Bernardino, H. S., Gonçalves, L. B., e Soares, S. S. (2021). Cheapest insertion and disruption of routes operators for solving multi-depot electric vehicle location routing problem with time windows and battery swapping via grasp and rvnd. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, p. 2133–2140. IEEE.
- Sbihi, A. e Eglese, R. W. (2010). Combinatorial optimization and green logistics. *Annals of Operations Research*, 175(1):159–175.
- Schiffer, M. e Walther, G. (2017). The electric location routing problem with time windows and partial recharging. *European Journal of Operational Research*, 260(3):995–1013.
- Toth, P. e Vigo, D. (2014). *Vehicle routing: problems, methods, and applications*, volume 18. Siam.
- Wu, T.-H., Low, C., e Bai, J.-W. (2002). Heuristic solutions to multi-depot location-routing problems. *Computers & Operations Research*, 29(10):1393–1415.
- Yang, J. e Sun, H. (2015). Battery swap station location-routing problem with capacitated electric vehicles. *Computers & Operations Research*, 55:217–232.
- Zarandi, M. H. F., Hemmati, A., e Davari, S. (2011). The multi-depot capacitated location-routing problem with fuzzy travel times. *Expert Systems with Applications*, 38(8):10075–10084.
- Zarandi, M. H. F., Hemmati, A., Davari, S., e Turksen, I. B. (2013). Capacitated location-routing problem with time windows under uncertainty. *Knowledge-Based Systems*, 37:480–489.

Tabela 3: Resultados experimentais do algoritmo RKGA-VND em comparação ao GRASP-CI-RVND [Portela et al., 2021] para 36 instâncias de 100, 144, 288 e 360 clientes.

INPUTS		GRASP-CI-RVND				RKGA-VND				$\Delta(\%)$
n	Instance	dp/vh/st	d^{best}	d^{avg}	t^{avg}	dp/vh/st	d^{best}	d^{avg}	t^{avg}	
100	c101-21	4/17/3	962.32	976.87	426.98	4/16/3	935.68	997.20	387.56	-2.77
	c109-21	4/16/3	932.39	942.86	461.06	3/14/3	892.49	920.67	430.05	-4.28
	c202-21	4/12/0	849.15	867.13	534.71	4/11/0	858.23	897.34	441.17	1.07
	c208-21	4/13/0	863.14	896.41	333.96	4/12/0	844.23	897.41	371.04	-2.19
	r101-21	4/27/8	1576.15	1594.66	323.67	4/25/10	1525.70	1567.83	510.53	-3.20
	r109-21	3/19/5	1262.12	1272.81	418.53	3/17/9	1201.46	1250.13	486.32	-4.81
	r206-21	3/12/0	976.78	995.24	495.62	3/10/0	943.76	978.71	512.95	-3.38
	r211-21	4/12/0	803.10	814.87	509.22	4/11/0	790.75	814.76	277.77	-1.54
	rc101-21	4/23/4	1452.05	1473.83	387.76	4/23/5	1431.11	1465.99	465.76	-1.44
	rc106-21	4/17/5	1219.83	1237.20	509.18	4/18/5	1250.77	1282.04	488.94	2.54
	rc203-21	4/13/0	935.33	946.58	551.84	4/11/1	905.19	963.12	409.74	-3.22
	rc208-21	4/9/0	801.21	834.36	332.96	4/10/0	811.27	855.75	297.15	1.26
	Avg.		1052.80	1071.07	440.46		1032.55	1074.25	423.25	-1.83
144	pr03-c	4/26/5	2617.52	2634.42	1233.48	4/23/7	2528.89	2646.54	2072.60	-3.39
	pr08-c	6/30/5	2371.30	2390.29	1181.35	6/26/7	2279.19	2317.62	1784.09	-3.88
	pr13-c	4/19/5	2225.73	2253.75	1363.14	4/19/6	2181.86	2275.30	1798.56	-1.97
	pr18-c	6/21/5	1961.92	2000.02	1152.86	6/20/6	1944.88	2002.58	1618.31	-0.87
	pr03-s	4/26/5	2610.25	2639.03	1211.01	4/25/6	2588.14	2644.84	2076.48	-0.85
	pr08-s	6/30/5	2381.84	2400.74	1140.41	6/25/6	2275.54	2305.71	1598.96	-4.46
	pr13-s	4/19/6	2243.95	2263.32	1340.57	4/19/6	2190.86	2254.36	1653.66	-2.37
	pr18-s	6/20/4	1961.37	1987.68	1147.66	6/19/7	1937.38	2021.92	1325.37	-1.22
	Avg.		2296.74	2321.16	1221.31		2247.98	2306.71	1663.62	-2.38
288	pr06-c	4/47/7	4129.96	4171.26	13235.66	4/45/11	4026.46	4096.94	35307.52	-2.51
	pr10-c	6/42/5	3886.80	3929.63	13453.73	6/43/5	3888.12	3949.65	33049.51	0.03
	pr16-c	4/34/6	3207.78	3273.02	16092.75	4/31/9	3268.31	3338.60	31781.54	1.89
	pr20-c	6/34/5	3251.91	3287.67	14426.64	6/31/6	3296.57	3379.71	27221.29	1.37
	pr06-s	4/44/8	4049.67	4086.38	12973.93	4/42/9	3907.88	4022.46	29305.14	-3.50
	pr10-s	6/42/5	3888.98	3933.64	13137.97	6/40/6	3822.07	3968.01	31248.42	-1.72
	pr16-s	4/32/7	3184.01	3223.07	15662.03	4/29/9	3196.48	3290.88	27462.85	0.39
	pr20-s	6/34/4	3215.03	3276.23	38702.59	6/34/7	3234.18	3332.03	27063.10	0.60
	Avg.		3601.77	3647.61	17210.66		3580.01	3672.29	30304.92	-0.43
360	pr11a-c	4/87/80	15073.50	16156.18	38283.24	4/53/47	9459.00	10249.27	88615.72	-37.25
	pr11b-c	4/41/13	6669.01	6856.49	39603.96	4/33/18	5752.42	6049.81	82846.89	-13.74
	pr17a-c	5/79/64	13088.70	13869.50	35523.37	5/54/41	9122.19	9522.25	80040.25	-30.30
	pr17b-c	6/38/6	5846.63	5976.38	36864.58	6/36/9	5686.71	5882.79	76726.26	-2.74
	pr11a-s	4/52/33	9075.42	9240.28	49087.09	4/47/32	8476.60	8743.52	83057.67	-6.60
	pr11b-s	4/36/14	6046.46	6129.62	39022.81	4/35/10	5653.56	5837.74	74498.17	-6.50
	pr17a-s	5/51/28	8551.83	8663.98	43030.16	5/45/32	7712.71	8078.48	83616.74	-9.81
	pr17b-s	6/40/5	5905.65	5966.90	34650.50	6/36/7	5593.81	5799.19	81532.77	-5.28
	Avg.		8782.15	9107.42	39508.21		7182.13	7520.38	81366.81	-14.03