

UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
INSTITUTO DE CIÊNCIAS EXATAS  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**Um GRASP-RVND para o Problema de  
Roteamento de Veículo Elétrico com Janelas  
de Tempo e Recargas Parciais**

**Filipe Augusto De Oliveira Leôncio**

JUIZ DE FORA  
NOVEMBRO, 2020

# Um GRASP-RVND para o Problema de Roteamento de Veículo Elétrico com Janelas de Tempo e Recargas Parciais

FILIPPE AUGUSTO DE OLIVEIRA LEÔNICO

Universidade Federal de Juiz de Fora  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Bacharelado em Ciência da Computação

Orientador: Luciana Brugiolo Gonçalves  
Coorientador: Lorenza Leão Oliveira Moreno

JUIZ DE FORA  
NOVEMBRO, 2020

UM GRASP-RVND PARA O PROBLEMA DE ROTEAMENTO  
DE VEÍCULO ELÉTRICO COM JANELAS DE TEMPO E  
RECARGAS PARCIAIS

Filipe Augusto De Oliveira Leôncio

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Luciana Brugiolo Gonçalves  
Doutora em Ciência da Computação - UFF

Lorenza Leão Oliveira Moreno  
Doutora em Informática - PUC-RIO

Juliana Mendes Nascente e Silva Zamith  
Doutora em Ciência da Computação - UFF

Stênio Sã Rosário Furtado Soares  
Doutor em Ciência da Computação - UFF

JUIZ DE FORA  
5 DE NOVEMBRO, 2020

*Aos meus amigos.*

*Aos pais, pelo apoio e sustento.*

## Resumo

O problema de roteamento de veículo elétrico com janelas de tempo e recargas parciais é uma variação do Problema de Roteamento de Veículos com Janela de Tempo, onde é permitido aos veículos elétricos pararem em estações de recarga para uma recarga parcial ou total da bateria. O objetivo deste problema é minimizar a quantidade de veículos necessária para atender os clientes e o tempo total gasto nas rotas dos veículos que corresponde a soma dos tempos de viagem, tempo de carregamento e tempos de espera. Este trabalho propõe o uso da heurística GRASP combinada com RVND para buscar boas soluções para o problema. Os resultados obtidos mostram que o método gera soluções com custos competitivos em relação à literatura e com tempo de processamento significativamente inferior.

**Palavras-chave:** Roteamento de veículo elétrico, recarga parcial da bateria, logística verde, GRASP Reativo.

## Abstract

The Electric Vehicle-Routing Problem with Time Windows and Partial Recharges is a VRPTW variant in which electric vehicles are allowed to stop in recharging stations to a fully or partial recharge. The goal is to minimize the amount of vehicles required to attend the customers and the total time spent in the vehicle's routes (i.e., travel times, charging times and waiting times). This work proposes a Reactive GRASP, composed by a constructive algorithm, solution refinements and a local search. Numerical results show that the method generates solutions relatively close, and sometimes better than the ones in the literature, spending significantly less processing time.

**Keywords:** Electric Vehicle Routing Problem, Green Logistic, Reactive GRASP.

## Agradecimentos

A todos os meus parentes, pelo encorajamento e apoio.

Aos professores Luciana, Lorenza e Stênio pela orientação, amizade e principalmente, pela paciência, sem a qual este trabalho não se realizaria.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

# Conteúdo

<b>Lista de Figuras</b>	<b>6</b>
<b>Lista de Tabelas</b>	<b>7</b>
<b>Lista de Abreviações</b>	<b>8</b>
<b>1 Introdução</b>	<b>9</b>
<b>2 Revisão Bibliográfica</b>	<b>11</b>
2.1 Descrição Formal do Problema . . . . .	11
2.2 Trabalhos Relacionados . . . . .	13
<b>3 Algoritmos Propostos</b>	<b>16</b>
3.1 Representação da Solução . . . . .	16
3.2 Pre-processamento . . . . .	17
3.3 Algoritmo Construtivo VIP . . . . .	18
3.4 RVND . . . . .	23
3.4.1 Movimentos . . . . .	24
3.4.2 Viabilização de Rotas . . . . .	25
3.4.3 Otimização de Rotas . . . . .	27
3.5 GRASP Reativo . . . . .	28
<b>4 Resultados Computacionais</b>	<b>30</b>
<b>5 Conclusões e Trabalhos Futuros</b>	<b>36</b>
<b>Bibliografia</b>	<b>37</b>



## Lista de Figuras

2.1	Exemplo de solução . . . . .	13
3.1	Representação de uma rota . . . . .	17
3.2	Refinamento 1 . . . . .	21
3.3	Refinamento 2 . . . . .	22
3.4	Refinamento 3 . . . . .	23
3.5	Movimento 1 - Realocação do cliente C1 em outra rota . . . . .	24
3.6	Movimento 2: Realocação de cliente C2 na mesma rota . . . . .	24
3.7	Movimento 3 - Troca do Cliente C3 com o Cliente C5 . . . . .	25
3.8	Viabilização 1: Realocação de Estação de Recarga. . . . .	26
3.9	Viabilização 2: Inserção de Nova Estação de Recarga. . . . .	26
3.10	Otimização: Realocação de estação de recarga . . . . .	27
3.11	Otimização: Remoção de estação de recarga . . . . .	27
4.1	Solução para a instância R202C15 . . . . .	33
4.2	Exemplo de Solução Para a Instância RC202C15 . . . . .	34

## Lista de Tabelas

4.1	Resultados para instâncias com 5 clientes . . . . .	32
4.2	Resultados para instâncias com 10 clientes . . . . .	32
4.3	Resultados para instâncias com 15 clientes . . . . .	33
4.4	Tabela de comparação das instâncias de 100 clientes . . . . .	35

## Lista de Abreviações

DCC	Departamento de Ciência da Computação
UFJF	Universidade Federal de Juiz de Fora
VRP	Vehicle Routing Problem (problema de roteamento de veículo)
VRPTW	Vehicle Routing Problem with Time Windows (problema de roteamento de veículo com janelas de tempo)
E-VRPTW	Electric Vehicle Routing Problem with Time Windows (problema de roteamento de veículo elétrico com janelas de tempo)
E-VRPTW-PR	Electric Vehicle Routing Problem with Time Windows and Partial Recharges (problema de roteamento de veículo elétrico com janelas de tempo e recargas parciais)

# 1 Introdução

A preservação do meio ambiente, preocupação crescente em todo o mundo, tem motivado mudanças significativas em vários setores da sociedade. Muitos governos têm investido em fontes de energia limpa e ampliado as regulamentações sobre emissões de poluentes e exploração de recursos não renováveis. Nas organizações do setor produtivo, o termo “logística verde” vem sendo usado para identificar processos logísticos que apresentam um equilíbrio entre a eficiência econômica e os impactos ambientais, considerando a sustentabilidade e o uso racional de recursos (SBIHI; EGGLESE, 2010).

Entre as diversas iniciativas governamentais com significativo impacto na área de logística verde, destaca-se a decisão de alguns países de banir o uso de veículos à combustão (DEKKER; BLOEMHOF; MALLIDIS, 2012). Em curto período de tempo é prevista a substituição destes veículos, com grande potencial de poluição atmosférica e sonora, por veículos elétricos, que podem ser recarregados através de energia renovável, não emitem gases poluentes e não causam poluição sonora. Tal mudança tem reflexos consideráveis no planejamento de transportes, uma vez que, atualmente, veículos elétricos ainda têm menor autonomia. O alto custo das baterias, a localização (ou escassez) de estações de recarga e o tempo expressivo necessário para esta operação são fatores que limitam as possibilidades de roteamento de frotas. Assim, dadas estas considerações, o planejamento das rotas de veículos elétricos deve considerar recargas ao longo do trajeto, prevendo o tempo desta operação e evitando grandes desvios, especialmente se o número de estações de recarga for escasso.

No meio acadêmico, boa parte dos problemas da logística de transporte são modelados através do problema de roteamento de veículos (VRP) e suas variações. No VRP, uma frota de veículos deve ser utilizada para atender demandas de clientes, a custo mínimo, de modo que cada cliente seja visitado exatamente uma vez e a capacidade dos veículos seja respeitada. Variações do problema consideram, por exemplo, janelas de tempo para atendimento dos clientes (VRPTW), frota heterogênea (HFVRP) ou possibilidade de entregas fracionadas (SDVRP). Com o advento dos veículos elétricos, surgiram

variantes do VRP que consideram a autonomia do veículo e abordam características específicas, como a existência de estações de troca de bateria ou de estações de recarga, com obrigatoriedade de recarga total ou a possibilidade de recarga parcial (SCHNEIDER; STENGER; GOEKE, 2014; BRUGLIERI et al., 2017).

Neste trabalho será tratado o problema de roteamento de veículo elétrico com janelas de tempo e recargas parciais (E-VRPTW-PR), que consiste em determinar rotas para veículos pertencentes a uma frota homogênea, de forma a atender todos os clientes, respeitando todas as restrições do problema e tendo como objetivo principal, minimizar a quantidade de veículos usados, então, o tempo total gasto nas rotas.

Para resolver esse problema, foi proposto um GRASP Reativo, onde na fase de construção se utiliza do conceito VIP, e a fase de busca local é feita através de um algoritmo do tipo RVND, usando 3 estruturas de vizinhança e alguns refinamentos para explorar o espaço de solução.

No próximo capítulo, Revisão Bibliográfica, é apresentada a descrição formal do problema, assim como trabalhos relacionados. Os algoritmos e as estruturas utilizadas, assim como a heurística proposta, são descritos no Capítulo 3. Os testes realizados e os resultados alcançados são discutidos no Capítulo 4, e as conclusões e trabalhos futuros são apresentados no Capítulo 5.

## 2 Revisão Bibliográfica

Neste capítulo são apresentados alguns trabalhos relacionados, bem como a descrição mais detalhada do problema de roteamento de veículo elétrico com janelas de tempo e recargas parciais, assim como a estratégia usada na literatura para resolver o problema.

### 2.1 Descrição Formal do Problema

O problema de roteamento elétrico com janelas de tempo e recargas parciais (E-VRPTW-PR), proposto por Bruglieri et al. (2017), consiste em determinar rotas de veículos elétricos para atendimento das demandas dos clientes, sendo que:

- toda a demanda do cliente precisa ser atendida, de uma só vez, cumprindo o tempo de serviço definido para cada cliente;
- o cliente deve ser visitado no período estipulado pela sua janela de tempo, sendo permitido ao veículo ficar parado, se necessário for, para aguardar o início do período de atendimento;
- a rota de cada veículo começa e termina no depósito e a soma das demandas dos clientes da rota não pode ultrapassar a capacidade de carga do veículo;
- a autonomia do veículo deve ser considerada e, em caso de necessidade, a bateria do veículo pode ser recarregada total ou parcialmente nas estações de recarga disponíveis.

O E-VRPTW-PR pode ser modelado sobre um grafo direcionado  $G = (V, A)$ , ponderado nos arcos, onde  $V = D \cup F$  é o conjunto de vértices, sendo  $D$  o conjunto de clientes e  $F$  o conjunto de estações de recarga e depósito. Para cada arco  $(i, j) \in A$ , são conhecidos a distância  $d_{ij}$  e o tempo de viagem  $t_{ij}$  entre os vértices  $i$  e  $j$ . Para cada cliente  $i \in D$ , são conhecidos a demanda  $q_i \geq 0$ , a janela de tempo  $[e_i, l_i]$  em que a entrega pode ser realizada, e o tempo de serviço  $s_i \geq 0$  necessário para a efetivação da entrega. São

conhecidos ainda o intervalo  $[e_0, l_0]$ , que especifica o período de tempo em que os veículos podem sair e entrar no depósito, que também pode ser utilizado como estação de recarga ao longo das rotas dos veículos. Quanto aos veículos, considera-se frota homogênea, e são conhecidos, a capacidade de carga de transporte  $C$ , a capacidade da bateria  $Q$ , o consumo de bateria  $r$  e a taxa de recarga  $g$ . Em qualquer estação de recarga, o tempo de recarga  $\rho$  é linearmente proporcional a  $g$ . O tempo total da rota de um veículo compreende o período entre o instante em que este deixa o depósito pela primeira vez até o instante em que retorna ao depósito pela última vez.

A Figura 2.1 ilustra uma solução ótima para uma instância do problema. No depósito  $D$ , são indicados a janela de tempo ( $JT$ ), bem como o instante de saída ( $TS$ ) e de chegada ( $TC$ ) de cada rota. Para cada cliente, são informados a janela de tempo ( $JT$ ), a bateria remanescente ( $BR$ ) e o instante de chegada ( $TC$ ). A cada parada em uma estação de recarga, são informados o tempo em que o veículo permanece parado para o abastecimento ( $TR$ ) e a bateria remanescente no veículo após a recarga ( $BR$ ). Neste exemplo simplificado, o peso de cada arco indica, simultaneamente, a distância entre os pontos, o tempo de viagem e a quantidade de bateria gasta para percorrê-lo. E ainda, o tempo de serviço em cada cliente é fixo e igual a 10 unidades e a capacidade da bateria nos veículos é 77. Em todas as estações, a bateria é recarregada a uma taxa de 2,5 unidades por instante de tempo. Considere que a capacidade do veículo é superior à soma de todas as demandas.

No exemplo, operações de recarga são necessárias em ambas as rotas, sendo que, na rota 2, duas paradas são realizadas na estação  $E1$ . Observe que, se a rota 2 tivesse uma única operação de recarga na estação  $E1$  após visitar o cliente  $C5$ , a autonomia do veículo seria respeitada, mas não seria possível alcançar o cliente  $C3$  antes do término de sua janela de tempo. Por isso, a bateria é totalmente recarregada antes da visita ao cliente  $C5$ , permitindo que o restante da bateria necessário para completar toda a rota seja recarregado rapidamente entre os clientes  $C5$  e  $C3$ .

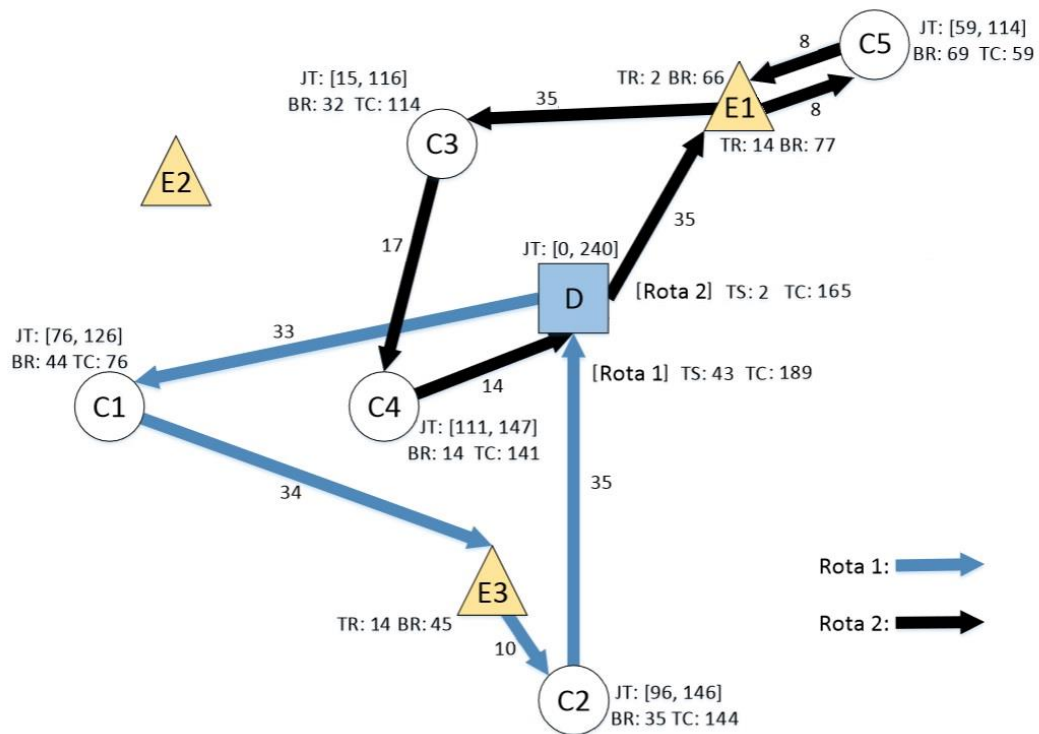


Figura 2.1: Exemplo de solução

## 2.2 Trabalhos Relacionados

Assim como o E-VRPTW (problema de roteamento de veículo elétrico com janelas de tempo), o problema de otimização E-VRPTW-PR, abordado nesse trabalho, é uma variação do VRPTW (problema de roteamento de veículo com janelas de tempo), cuja complexidade é NP-difícil. Assim, não se conhece algoritmo que resolva o problema na otimalidade em tempo polinomial, o que desencoraja o uso de abordagens exatas para resolver instâncias de tamanho real. Nestas condições, o uso de metaheurísticas puras ou combinadas com métodos exatos é comumente encontrado em trabalhos da literatura.

O G-VRP (Problema de Roteamento de Veículo Verde), em uma variação onde a capacidade dos veículos e as janelas de tempo não são consideradas, foi solucionado por Erdoğan e Miller-Hooks (2012), onde apresentaram um MILP (Mixed-Integer Linear Programming) em conjunto com várias técnicas para obter uma solução que minimiza a distância total percorrida, enquanto adicionando paradas para o reabastecimento dos AFVs (Veículos de Combustível Alternativo) nas estações de combustível alternativo. Em



uma outra variação do G-VRP, em que são consideradas múltiplas tecnologias e recargas parciais, onde recargas parciais da bateria podem ser feitas durante a noite no depósito, e elas também podem ser efetuadas com diferentes tecnologias sendo que cada uma delas tem um tempo e um custo associados, foi tratado por Felipe et al. (2014), em que propuseram um algoritmo construtivo baseado em um método de geração guloso, uma busca local determinística e um simulated annealing. Já Ćirović, Pamučar e Božanić (2014) lidaram com o G-VRP com uma frota heterogênea composta por veículos ambientalmente amigáveis e não amigáveis, no entanto, quando a rota é definida, esses veículos são considerados separadamente, sendo que para a solução do problema, foi usado um modelo neuro-fuzzy.

O E-VRPTW, introduzido em Schneider, Stenger e Goeke (2014), lida com a possibilidade de recarga em qualquer estação disponível, sendo que a operação de recarga sempre completa toda a carga da bateria e, por isso, o tempo de recarga varia com o nível da bateria do veículo ao chegar na estação. Nesse caso, dois objetivos hierárquicos são considerados para minimização: primeiro, o número de veículos presentes na solução e, em seguida, a distância total percorrida nas rotas. O problema foi tratado usando algoritmo exato sobre um modelo de programação linear inteira mista, combinado com uma metaheurística híbrida composta por um VNS (Variable Neighborhood Search) com um método de TS (Tabu Search) para a fase de intensificação do VNS.

O E-VRPTW-PR foi abordado em Keskin e Çatay (2016), com o objetivo de minimizar a distância total de viagem dos veículos. A proposta dos autores integra quatro heurísticas envolvendo: remoção de cliente, inserção de cliente, remoção de estação e inserção de estação. Em Hiermann et al. (2016), uma frota heterogênea de veículos elétricos é utilizada, cada um com um custo associado, sendo o objetivo minimizar os custos de aquisição e a distância total percorrida. O problema é resolvido pelo uso de um algoritmo de branch-and-price e o trabalho ainda propõe uma heurística híbrida, que combina uma busca adaptativa na vizinhança com busca local e procedimento de rotulação.

Uma variação do problema de roteamento de veículos elétricos que considera, além de frota heterogênea, o tempo de recarga e um tempo de viagem variável do cliente foi tratada em Shao et al. (2017). Neste trabalho, os autores propõem uma abordagem evolucionista que retorna o conjunto de rotas indicando o horário de saída do depósito

para cada veículo e o plano de recarga dos mesmos. Além do algoritmo genético, utiliza ainda um algoritmo de caminho mínimo baseado no algoritmo de Dijkstra para obter caminhos mais curtos entre os vértices ao longo das rotas.

Ainda sobre o E-VRPTW-PR, uma abordagem híbrida foi proposta em Bruglieri et al. (2017). A abordagem integra três fases para a obtenção da solução do problema:

- Fase 1: o algoritmo obtém uma solução viável  $S^1$  através da solução de um modelo de programação inteira e linear mista usado em Erdoğan e Miller-Hooks (2012) para o problema do roteamento de veículo verde (GVRP), com a inclusão de janelas de tempo, e restrições de capacidade, que acaba sendo um problema mais fácil de se resolver, com isso, é obtido o tempo total gasto nas rotas, já minimizado.
- Fase 2: a solução  $S^1$  é dada como entrada para o modelo do E-VRPTW-PR usado em Bruglieri et al. (2015b), que tem como objetivo otimizar o tempo de saída do depósito no início das rotas, e como consequência, o tempo de chegada nos nós, e também o nível de bateria de chegada em cada estação de recarga, assim como o nível de bateria ao chegar em cada cliente, gerando a solução  $S^2$ .
- Fase 3: a solução  $S^2$  é o ponto inicial para o VNSB (*variable neighborhood search e local branching*) usado em Bruglieri et al. (2015a), onde o algoritmo VNS adiciona restrições lineares ao problema original para a troca de vizinhanças. Um algoritmo de *local branching* é usado adicionando restrições lineares ao modelo de programação inteira e linear mista considerando a distância de Hamming entre diferentes soluções, baseando-se nas vizinhanças (FISCHETTI; LODI, 2003). Através de um LNS (*Large Neighborhood Search*), as vizinhanças são analisadas sequencialmente. Cada vizinhança é definida ao se adicionar uma restrição ao modelo, onde a distância de Hamming entre a solução da vizinhança e a atual deve ser menor ou igual a um raio pré-determinado. Modelo este que é resolvido pelo CPLEX. Então, uma fase de perturbação avalia as vizinhanças desarticuladas, e o modelo resultante é resolvido pelo CPLEX. A primeira solução viável encontrada é mantida como atual, e dela, o LNS recomeça. A fase 3, assim como toda a meta-heurística, termina quando um dado tempo de CPU é alcançado.

### 3 Algoritmos Propostos

A metaheurística GRASP, *Greedy Randomized Adaptive Search Procedure*, proposta por Feo e Resende (1995), é um algoritmo iterativo em que cada laço envolve duas fases: construção e busca local. Na fase de construção é gerada uma solução através de um algoritmo guloso, randomizado e adaptativo. Em seguida, a solução é submetida a um procedimento de busca local. A abordagem proposta neste trabalho para o E-VRPTW-PR consiste em um algoritmo GRASP Reativo (PRAIS; RIBEIRO, 2000), onde o parâmetro de aleatoriedade do algoritmo construtivo é ajustado durante a execução do GRASP, combinado com um algoritmo RVND (*Random Variable Neighborhood Descent* (SUBRAMANIAN et al., 2010)) que combina diferentes vizinhanças na busca local. Os detalhes da estratégia proposta são apresentados nas próximas seções.

Uma vez que o objetivo principal do problema é minimizar o número de veículos, o algoritmo proposto considera este aspecto ao longo da busca pela melhor solução. Como esse fator não é considerado na função objetivo apresentada por Bruglieri et al. (2017), neste trabalho foi utilizada uma função auxiliar para guiar o processo de busca onde foi acrescentada uma segunda parcela incorporado um custo  $M$  a cada um dos  $k$  veículos utilizados. Usando esta função, apresentada na Equação 3.1, é possível comparar soluções com número diferente de veículos.

$$\min \sum_{i \in V} \sum_{j \in V} t_{ij} x_{ij} + M \times k \quad (3.1)$$

#### 3.1 Representação da Solução

Para representar uma solução para o E-VRPTW-PR é utilizada uma lista de veículos. Para cada veículo há uma lista de nós, equivalente à sua rota. Esses nós podem ser de dois tipos, clientes ou estações de recarga. Numa dada rota são armazenadas algumas informações para cada nó, como apresentadas a seguir:

- Clientes: tempo de abertura da janela (TA); tempo de fechamento da janela (TF);

tempo de chegada (TC); tempo de serviço (TSe), que corresponde ao tempo de serviço do cliente; tempo de início de atendimento (TIa), definido como o maior valor entre TA e TC; tempo de saída (TS), dado como TIa+TSe; tempo de espera (TE) caso o tempo de chegada seja menor que o de abertura, definido como TA-TC; tamanho da janela (TJ), sendo o TF-TA; tempo de folga na rota (FR), sendo o maior atraso que pode se ter naquele ponto sem inviabilizar o restante da rota pelo fato da perda de alguma janela de tempo, e bateria remanescente (BR), que corresponde ao nível de bateria do veículo no momento em que ele sai do cliente.

- Estações de recarga: Tempo de chegada (TC); tempo de saída (TS); tempo em que o veículo ficou recarregando sua bateria (TSe); tempo de folga na rota (FR); bateria remanescente (BR) e bateria que foi carregada (BC).

A Figura 3.1 ilustra esta representação para a rota de um veículo.

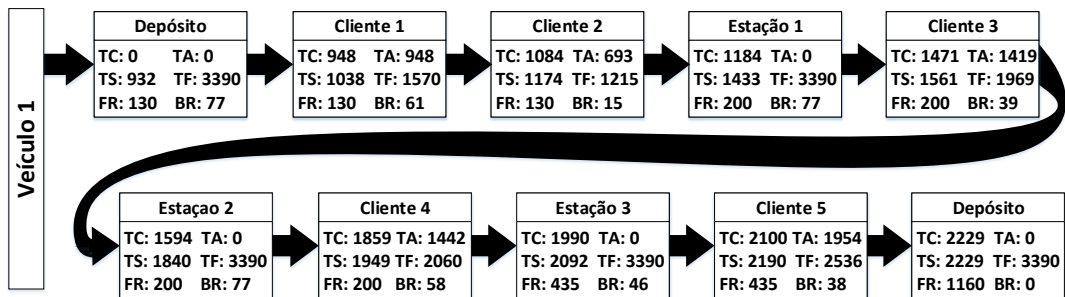


Figura 3.1: Representação de uma rota

## 3.2 Pre-processamento

A fim de poupar tempo na execução, foram criadas duas estruturas de pré-processamento, contendo informações que são utilizadas ao longo de todo o algoritmo.

- Estrutura 1: Matriz de tamanho  $N \times N \times R$  (onde  $N$  equivale ao número de nós do grafo e  $R$  a quantidade de estações de recarga a serem consideradas) onde, para cada par de nós, tem-se as  $R$  estações mais próximas dos dois nós, armazenando informações como tempo e bateria necessários para o desvio.

- Estrutura 2: Vetor de tamanho  $N$ , onde em cada posição contém a estação mais próxima daquele nó, junto com sua distância e bateria necessária, assim como a distancia e bateria até o deposito.

### 3.3 Algoritmo Construtivo VIP

Dadas as restrições do E-VRPTW-PR, obter uma solução viável por heurísticas construtivas é uma tarefa árdua. A estratégia proposta é baseada na construção paralela das rotas, onde as rotas são definidas simultaneamente.

Para cada par  $(c, r)$ , onde  $c$  é um cliente e  $r$  é uma rota, é calculado um custo associado a inserção do cliente  $c$  na rota  $r$ . A partir destes custos, a cada iteração do algoritmo, são selecionados o cliente ( $c$ ) e a rota ( $r$ ) em que este será inserido.

Durante o processo de construção, além do custo associado a cada par  $(c, r)$ , considera-se também a urgência no atendimento ao cliente, de forma a dar maior prioridade àqueles clientes com menor folga em relação aos demais. Para indicar esta prioridade, foi definido o conceito de *Cliente VIP*. Para favorecer a viabilidade no que se refere à janela de tempo de atendimento, o conceito de Cliente VIP visa priorizar os clientes ainda não inseridos na solução, cujas janelas de tempo de atendimento estejam mais restritivas. Para tanto, a cada iteração do algoritmo construtivo, são considerados Clientes VIP em relação a uma rota  $j$  todos aqueles que possuem suas janelas de tempo relativas menores do que  $1/m$  da média do tamanho das janelas relativas de todos os clientes, onde  $m$  é um inteiro escolhido pelo algoritmo. O conceito janela relativa de um cliente equivale ao tamanho da janela, caso o tempo de saída do último nó visitado na rota  $j$ , dito *tempo atual da rota*, seja menor que o tempo de abertura da janela deste cliente, ou a diferença entre o tempo de fechamento da sua janela e o tempo de saída do último nó, caso o tempo atual da rota seja maior que a abertura da janela. Note que, quanto maior o valor de  $m$ , menor será a quantidade de clientes VIP, visto que menor será o tamanho da janela necessária para o cliente ser classificado como tal. Vale observar que, durante a construção, a inserção de clientes ocorre sempre no final das rotas.

O construtivo proposto é apresentado no Algoritmo 1, onde o parâmetro  $\alpha$  indica o grau de aleatoriedade do algoritmo e  $t$  indica o percentual das possíveis inserções a

serem consideradas a cada operação de inserção.

---

**Algoritmo 1:** Construtivo VIP
 

---

**Input:**  $\alpha, t$   
**Output:** Solução Inicial S

```

1  $m \leftarrow \text{rand}() \bmod 5 + 2;$ 
2  $qtd \leftarrow \text{SomaDemandasClientes} / \text{capacidadeVeiculo};$ 
3  $S \leftarrow \text{inicializa\_Solucao}(qtd);$ 
4 for cada veiculo  $j$  do
5    $\text{inicializa}(\text{listaVIP}_j, m);$ 
6   for cada cliente  $C \in \text{listaVIP}_j$  do
7      $\text{adiciona a } LI_j \text{ aqueles que podem ser inseridos antes de } C \text{ e o}$ 
8      $\text{próprio } C;$ 
9   end for
10   $\text{Ordena } LI_j (C1, C2, C3);$ 
11 end for
12 while  $qtd\text{ClientesVisitados} < \text{numeroClientes}$  do
13    $LMI \leftarrow \text{melhoresInsercoes}(t);$ 
14   if  $LMI$  vazio then
15      $\text{Adiciona novo veículo};$ 
16      $LMI \leftarrow \text{melhoresInsercoes}(t);$ 
17   end if
18    $LMI \leftarrow \text{ordena}(LMI, C1, C2);$ 
19    $(c, r) \leftarrow \text{sorteio}(LMI, \alpha);$ 
20    $S \leftarrow \text{Insere cliente } c \text{ na rota } r;$ 
21   if  $c \in \text{listaVIP}_r$  then
22     for Cada veiculo  $s$  diferente de  $r$  do
23        $\text{Atualiza listaVIP}_s \text{ e } LI_s \text{ e ordena } LI_s;$ 
24     end for
25   end if
26    $\text{Atualiza ListaVIP}_r \text{ e } LI_r \text{ e ordena } LI_r;$ 
27 end while

```

---

Neste algoritmo, inicialmente é definido o valor de  $m$ , utilizado para indicar o status VIP dos clientes (linha 1). Em seguida, na linha 2, é calculado o número mínimo de veículos necessário para atender as demandas dos clientes,  $qtd$ . Definido o valor de  $qtd$ , na linha 3 é criada uma solução  $S$  com uma rota para cada um dos veículos. Neste passo, cada rota é inicializada com o depósito.

Em seguida, para cada veículo  $j$  é criada sua lista de inserções  $LI_j$  contendo os clientes candidatos a inserção na rota  $j$  (linhas 4 a 10). Para tal, na linha 5 é criada a  $\text{listaVIP}_j$  contendo todos os clientes VIPs em relação ao veículo  $j$ . Logo após definir os clientes VIPs é definida a Lista de Inserções para o veículo  $j$  ( $LI_j$ ). Em  $LI_j$  são inseridos todos os clientes VIPs, bem como os demais clientes que podem ser atendidos antes de um

cliente VIP, mantendo a garantia de que o cliente VIP poderá ser atendido na sequência. Se a lista VIP de  $j$  estiver vazia, a Lista de Inserções  $LI_j$  será composta por todos aqueles clientes que podem ser inseridos na rota do veículo  $j$ .

Após construída,  $LI_j$  é ordenada (linha 9), considerando os critérios: tamanho da janela relativa (C1), tempo de espera (C2) e custo final de inserção, que inclui tempo de viagem, tempo de recarga (se houver) e tempo de espera (C3). Para ordenação é determinado um custo total, calculado pela combinação linear destes valores:  $\kappa * C1 + \rho * C2 + \phi * C3$ .

Depois, enquanto houver cliente a ser atendido (linha 11), é criada uma lista de melhores inserções LMI (linha 12), contendo as  $t$  melhores inserções para cada veículo, obtidas a partir das LIs. Em LMI são incluídos pares  $(c,r)$  indicando o cliente  $c$  e a rota  $r$  em que deve ser inserido. Caso a lista LMI esteja vazia (linha 13), é detectada a necessidade de incluir um novo veículo na solução, dado que os clientes ainda não atendidos não podem ser inseridos nas rotas dos veículos atualmente presentes na solução. A rota do novo veículo é inicializada da mesma maneira que as demais (linha 14) e a LMI é atualizada (linha 15). Na linha 17,  $LMI$  é ordenada de acordo com o tempo de espera e o custo final de inserção. O par  $(c, r)$  é escolhido aleatoriamente entre os  $\alpha$  por cento primeiros pares de LMI (linha 18).

A inserção do cliente é feita sempre no final da rota, e pode ser de dois tipos: Inserção Tipo 1 - apenas o cliente é inserido, caso a bateria seja suficiente para atender o cliente e chegar à estação de recarga mais próxima ou ao depósito; ou Inserção Tipo 2 - quando a bateria não é suficiente, é inserida uma estação de recarga, e então o cliente. Nesta etapa da abordagem proposta, mesmo o problema permitindo recarga parcial, sempre que uma estação de recarga é inserida na solução, é dado tempo suficiente para recarga total da bateria.

Em seguida, nas linhas 20 a 25, as listas VIP e listas de inserções dos veículos são atualizadas, e as LI são ordenadas.

Uma vez gerada uma solução, antes de submetê-la à fase de busca local, três estratégias de refinamento são aplicadas, sendo elas:

- R1 - Atrasar saída do depósito: Consiste em tentar atrasar a saída do veículo

do depósito, reduzindo o tempo de espera em alguns clientes. Para isso, para cada cliente na rota foi calculado qual o tempo mais tarde em que ele poderia ser visitado, sem perder a janela de nenhum outro cliente no restante da rota. Na Figura 3.2 pode-se observar que antes do refinamento se tem uma folga rota (FR) de 303 nos dois primeiros nós, e após o refinamento, ocorrem mudanças nos tempos de saída, chegada e folga rota da maioria, zerando o tempo de espera no cliente 1, e diminuindo ao máximo o tempo de espera do cliente 2.

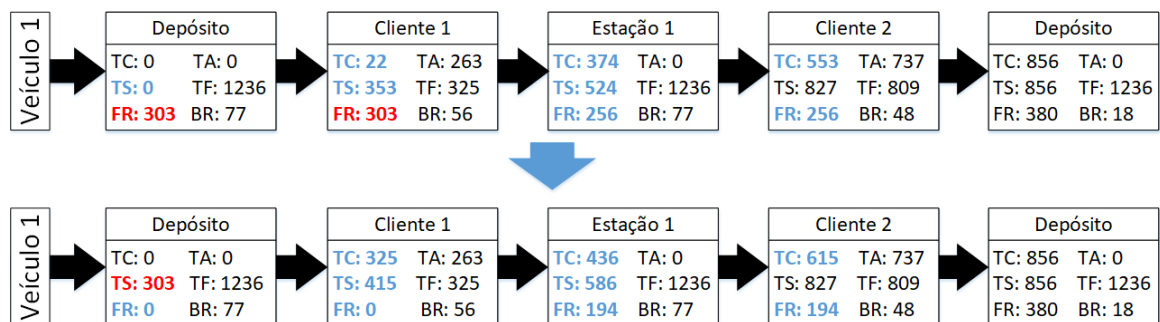


Figura 3.2: Refinamento 1

- R2 - Redução última recarga: Consiste em tentar fazer com que a bateria do veículo, no retorno ao depósito, seja zero. Isto é feito diminuindo a última recarga feita por aquele veículo, recarregando somente o necessário para que ele complete a sua rota. Na Figura 3.3 pode-se observar que, antes do refinamento, a bateria de saída na Estação 1 era 77, referente a uma recarga total, após o refinamento, a bateria de saída reduz para 59, mudando assim, o tempo em que o veículo passa carregando na estação, e portanto, o tempo de chegada no próximo cliente.



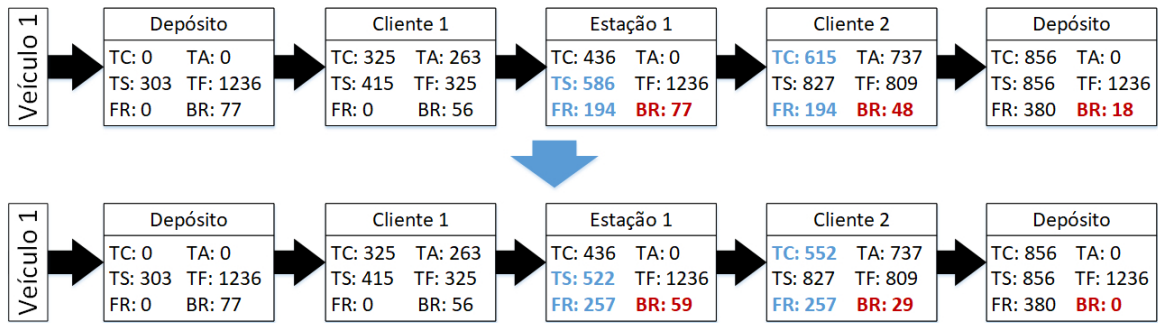


Figura 3.3: Refinamento 2

- R3 - Transformar tempo de espera em tempo de recarga: Dadas duas estações, onde uma destas possui alguma folga, busca-se a melhorar distribuição do tempo necessário de recarga a fim de transformar o tempo em espera que gerou a folga na estação em tempo em recarga. Na Figura 3.4 pode ser observar que antes do refinamento, são gastas 60 unidades de tempo para completar a recarga da bateria, fazendo com que o veículo chegue na estação 1 com 20 de bateria, onde são gastos mais 25 unidades de tempo para o veículo sair com 45 de bateria, sendo o suficiente para chegar até o depósito, porém, no cliente C2 tem-se tempo de espera igual a 25. Visando acabar com esse tempo, é retirado 20 unidades de tempo da estação E2 e transferido para a estação E1, tornando possível o atraso na saída do depósito em 20 unidades de tempo, que antes não era possível porque o cliente C3 não tinha folga nenhuma pois já estava sendo atendido no limite de sua janela, reduzindo o custo final da rota em 20.

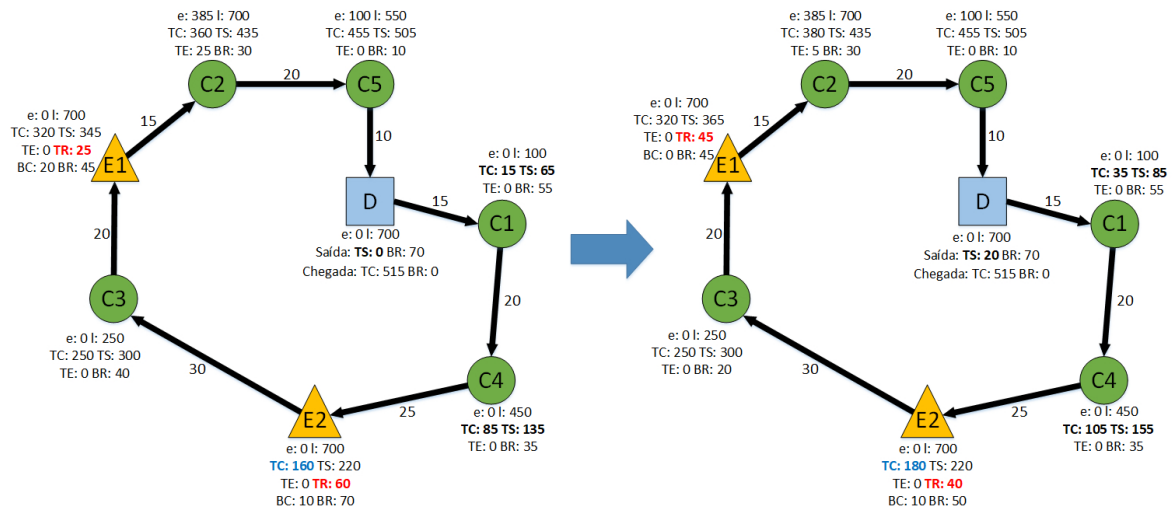


Figura 3.4: Refinamento 3

Essas funções são executadas de forma que, se uma gera mudança na solução, as outras são executadas novamente, ou seja, o refinamento só acaba quando nenhuma das três funções consegue melhoria. Nesta etapa, a recarga parcial é considerada para refinamento da solução.

### 3.4 RVND

Variable Neighborhood Descend (VND) é uma estratégia de melhoramento local que é comumente usada como subordinada em um Variable Neighborhood Search (HANSEN; MLADENOVIĆ, 1999) ou outras meta-heurísticas. No VND, são empregadas diferentes estruturas de vizinhança  $N_1, \dots, N_n$ , que são exploradas a partir de trocas sistemáticas. A solução final representa um ótimo local para todas as vizinhanças. Um bom desempenho do VND depende de uma boa escolha na ordem em que serão aplicadas as estruturas de vizinhança. Tentando resolver esse problema, surgiu o RVND, que aplica, a cada chamada do método, as estruturas de vizinhança em uma ordem aleatória (SUBRAMANIAN et al., 2010; HU; RAIDL, 2006).

### 3.4.1 Movimentos

Para explorar o espaço de soluções do E-VRPTW-PR, três diferentes movimentos foram utilizados. Cada movimento define uma vizinhança  $N(\cdot)$ , e são descritos a seguir.

- $N_1(s)$  - **Realocação de cliente em outra rota:** Consiste em retirar um cliente de uma rota e tentar inserí-lo em outra, no melhor ponto possível, como ilustra a Figura 3.5

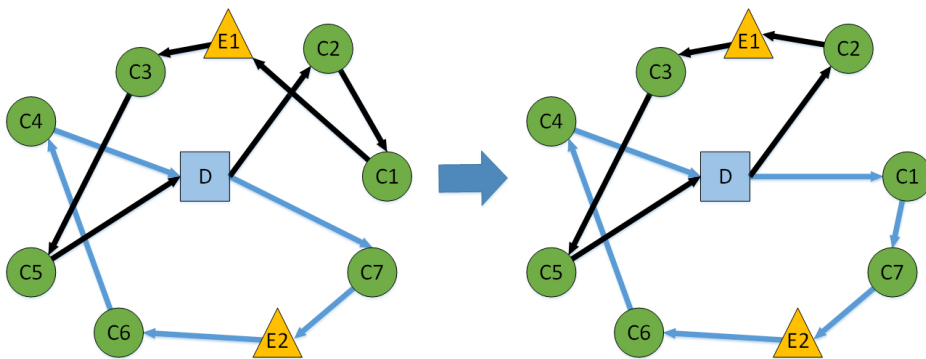


Figura 3.5: Movimento 1 - Realocação do cliente C1 em outra rota

- $N_2(s)$  - **Realocação de cliente na mesma rota:** Consiste em trocar a posição de um cliente dentro da mesma rota e ajustar as escolhas de recarga, a fim de gerar ganho no tempo da solução, como exemplificado na Figura 3.6.

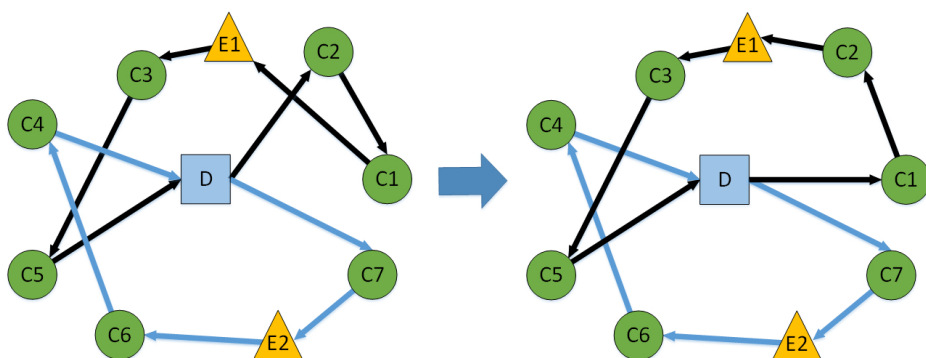


Figura 3.6: Movimento 2: Realocação de cliente C2 na mesma rota

- $N_3(s)$  - **Troca de Clientes Entre Rotas:** Dados dois clientes de rotas distintas, este movimento consiste em trocar a posição destes, ajustando as escolhas de recarga de bateria, como mostra a Figura 3.7

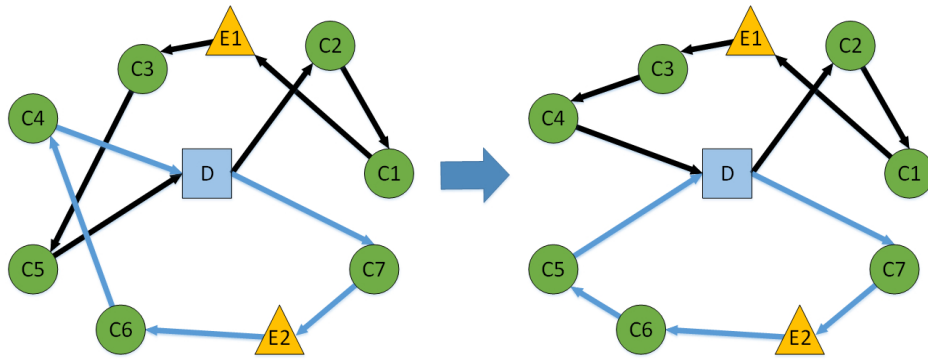


Figura 3.7: Movimento 3 - Troca do Cliente C3 com o Cliente C5

Cada um destes movimentos, quando aplicado em relação a uma solução  $s$ , determina as soluções que pertencem à vizinhança de  $s$ , representada por  $N_i(s)$ . Como estratégia de exploração da vizinhança, neste trabalho optou-se por utilizar o melhor aprimorante, ou seja, após explorar toda a vizinhança de  $s$ , é retornada a melhor solução  $s'$  encontrada, caso ela seja melhor que a solução de entrada  $s$ .

### 3.4.2 Viabilização de Rotas

No processo de determinação das soluções que compõem  $N_i(s)$ , após cada movimento realizado é verificada a viabilidade das rotas no que se refere às restrições relacionadas à janela de tempo e limite de carga da bateria. Para tratar uma eventual inviabilidade, a rota é percorrida até que seja identificado um ponto em que alguma das restrições não é atendida. Então, tenta-se viabilizar o trecho do começo até o ponto em questão. Neste cenário, dois tipos ajustes podem ser feitos já considerando uma possível recarga parcial:

- **Realocação de Estações de Recarga:** Retira-se uma estação intermediária anterior ao ponto de inviabilidade, na tentativa de realocá-la em outro ponto. Neste processo, entre as possíveis formas de sanar a inviabilidade, opta-se pela de menor custo. Este ajuste é ilustrado na Figura 3.8, onde foi detectado que a bateria do veículo não é suficiente para alcançar o cliente  $C3$ . Para corrigir este problema, a estação  $E2$  é realocada, saindo do caminho entre os clientes  $C1$  e  $C2$ , e sendo reinserida entre o depósito  $D$  e o cliente  $C1$ .

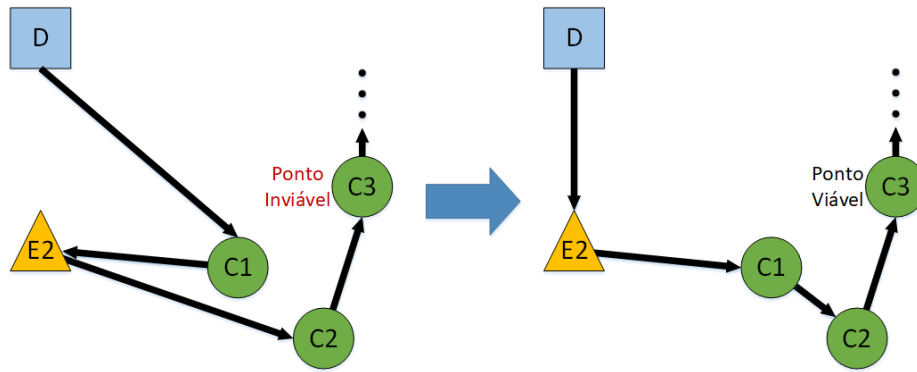


Figura 3.8: Viabilização 1: Realocação de Estação de Recarga.

- Inserção de Nova Estação de Recarga: Tenta-se inserir uma nova estação de recarga em todos os pontos anteriores à inviabilidade, escolhendo a inserção que gera o menor acréscimo de custo na solução. Esta viabilização é exemplificada na Figura 3.9, onde foi necessário incluir uma nova estação antes do retorno do veículo ao depósito.

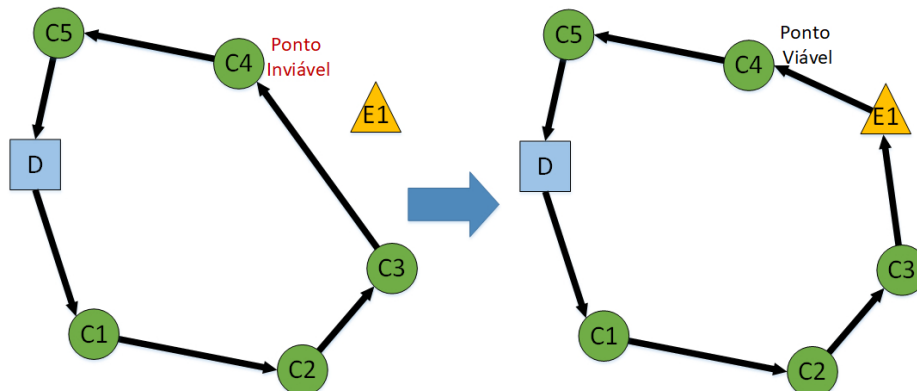


Figura 3.9: Viabilização 2: Inserção de Nova Estação de Recarga.

Esses ajustes de viabilização são executados em sequência, caso o primeiro não torne o trecho viável, o segundo é executado; com exceção do caso em que não existam estações de recarga no trecho do depósito até o ponto de inviabilidade, onde a única alternativa é inserir uma nova estação. Caso não seja possível a viabilização, o movimento que gerou a rota em questão é considerado inviável e a solução é descartada.

### 3.4.3 Otimização de Rotas

Após a viabilização das rotas, um processo simples de otimização é executado para verificar a possibilidade de ajustes na solução para reduzir o tempo total das rotas.

Neste processo, uma rota é dividida em trechos compostos por 3 estações de recargas consecutivas (início, meio e fim) e os clientes entre elas. Cada trecho é submetido a uma função que verifica a possibilidade de reorganização das estações, a fim de melhorar a qualidade da solução, onde dois casos, representados pelas Figuras 3.10 e 3.11. Enquanto a Figura 3.10 ilustra um contexto onde uma estação foi realocada, na Figura 3.11 é apresentado um cenário onde foi possível a remoção de uma estação.

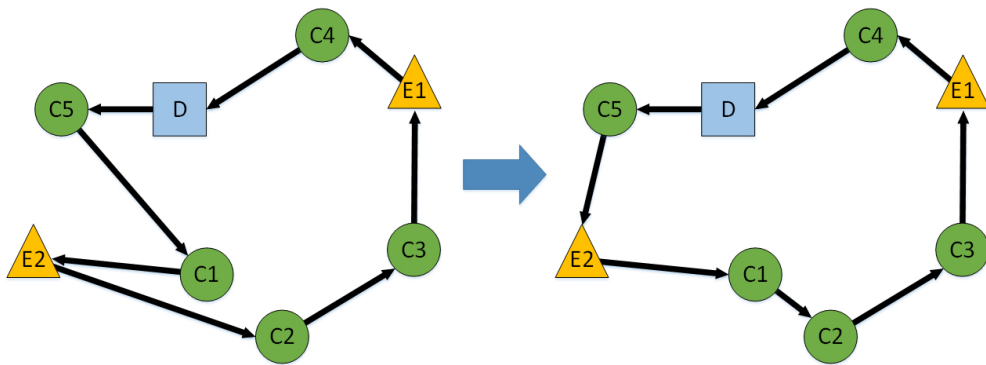


Figura 3.10: Otimização: Realocação de estação de recarga

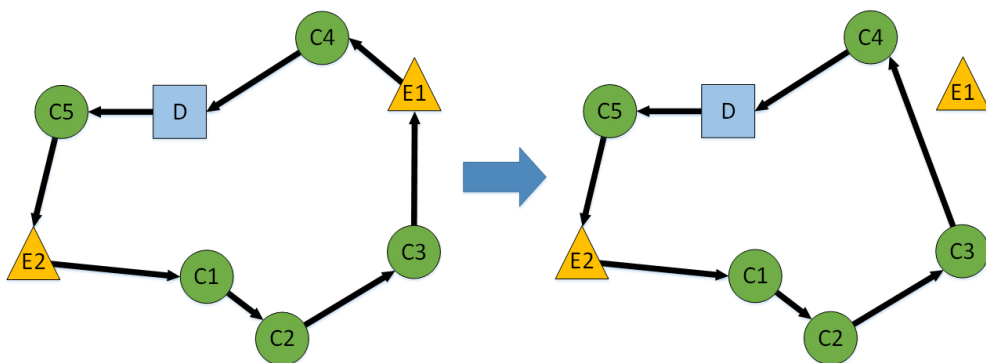


Figura 3.11: Otimização: Remoção de estação de recarga

Como estratégia de combinação das três diferentes vizinhanças adotadas, optou-se pelo RVND, como apresentado no Algoritmo 2. A solução resultante deste algoritmo é um ótimo local em relação às três vizinhanças utilizadas.

**Algoritmo 2: RVND**


---

**Input:**  $r$  vizinhança em ordem randômica:  $N_1, N_2$  e  $N_3$ .  
**Input:** solução inicial  $s$   
**Output:**  $s$

```

1  $k \leftarrow 1$ ;
2 while  $k \leq r$  do
3   encontrar a melhor solução  $s'$  de  $N_k(s)$ ;
4   if  $s'$  melhor que  $s$  then
5      $s \leftarrow s'$ ;
6      $k \leftarrow 1$ ;
7   else
8      $k \leftarrow k + 1$ ;
9   end if
10 end while

```

---

### 3.5 GRASP Reativo

A metaheurística GRASP se destaca por ser uma estratégia simples de se implementar, eficiente para diversos tipos de problemas (RESENDE; RIBEIRO, 2010) e por possuir apenas um parâmetro além do critério de parada: o parâmetro  $\alpha$  responsável por ajustar o grau de aleatoriedade do algoritmo de construção. De acordo com estes mesmos autores, Resende e Ribeiro (2010), usar um único valor para o parâmetro  $\alpha$  pode dificultar o processo de busca, além do mais, diferentes valores para  $\alpha$  podem ser adequados para diferentes instâncias. Nesta direção, foi proposto o GRASP Reativo, uma extensão do GRASP onde o valor de  $\alpha$  é ajustado dinamicamente durante a execução do algoritmo de acordo com a qualidade das soluções obtidas durante o processo de busca.

No Algoritmo 3, é possível verificar as etapas da estratégia proposta. Observe que, como parâmetros, tem-se o conjunto com  $m$  possíveis valores para  $\alpha$  e  $maxIter$  e  $maxISM$  associados ao critério de parada do algoritmo. Enquanto  $maxIter$  corresponde ao número máximo de iterações,  $maxISM$  indica o número máximo de iterações sem melhora permitido pelo algoritmo.

**Algoritmo 3:** GRASP Reativo

---

**Input:**  $m$  possíveis valores para  $\alpha$ ,  $\text{maxIter}$ ,  $\text{maxISM}$ .

```

1  inicializaProbabilidades( $P$ ,  $m$ );
2   $S^* \leftarrow \emptyset$ ;
3   $\text{itSemMelhora} \leftarrow 0$ ;
4  for  $i \leftarrow 1 \dots \text{maxIter}$  ou  $\text{itSemMelhora} < \text{maxISM}$  do
5       $\alpha \leftarrow \text{sorteio}(P)$ ;
6       $S \leftarrow \text{Construtivo VIP}(\alpha)$ ;
7       $S' \leftarrow \text{Refinamentos}(S)$ ;
8       $S_f \leftarrow \text{RVND}(S')$ ;
9      if  $\text{custo}(S_f) < \text{custo}(S^*)$  then
10          $S^* \leftarrow S_f$ ;
11          $\text{itSemMelhora} \leftarrow 0$ ;
12     else
13          $\text{itSemMelhora}++$ ;
14     end if
15     if  $i \bmod t\text{Bloco} = 0$  then
16          $\text{ajustaProbabilidades}(P)$ ;
17     end if
18 end for

```

---

Na linha 1, o vetor  $P$  é inicializado com a mesma probabilidade  $1/m$ , associada a cada valor candidato ao parâmetro  $\alpha$ . A cada iteração do GRASP (linhas 4-18), o primeiro passo é determinar o valor de  $\alpha$  a ser utilizado na iteração corrente, a chance de escolha de cada valor é indicada em  $P$ . Definido o valor de  $\alpha$ , na linha seguinte uma solução é construída. Os três refinamentos apresentados na Seção 3.1 são aplicados na linha 7. Em seguida, a solução obtida é submetida ao RVND. Caso a solução retornada pela estratégia de busca local seja melhor que  $S^*$ , esta é atualizada na linha 10 e a variável que contabiliza iterações sem melhora é zerada na linha 11. Caso contrário, incrementa-se a variável  $\text{itSemMelhora}$  na linha 13. A cada conjunto de  $t\text{Bloco}$  iterações, as probabilidades de escolha dos valores de  $\alpha$  são ajustadas considerando-se a média das soluções obtidas por cada  $\alpha$  (linha 16), esta atualização é explicada em Resende e Ribeiro (2010).



## 4 Resultados Computacionais

Para avaliar a eficiência dos algoritmos propostos, testes computacionais foram realizados sobre o conjunto de 36 instâncias utilizadas em Bruglieri et al. (2017) para o EVRPTW-PR, que foram cedidas pelos autores de Schneider, Stenger e Goeke (2014). Na definição das instâncias, o número de clientes foi fixado em grupos com 5, 10 e 15 clientes, cuja distribuição geográfica deu-se de forma randômica, baseada em clusters, ou combinando estas duas formas, gerando os grupos C, R e RC, respectivamente. Dentro de cada grupo, diferentes instâncias foram criadas conforme combinações de valores de capacidade de bateria, capacidade de carga e taxa de recarga de bateria  $g$ . Para maiores detalhes sobre as instâncias, veja (SCHNEIDER; STENGER; GOEKE, 2014).

Os algoritmos foram desenvolvidos na linguagem C++ e executados em um PC AMD FX 6300 de 3.5Ghz, com 8 GB de memória, executando Linux Ubuntu 18.04.3 LTS. Cada instância foi submetida a 30 execuções independentes e o melhor resultado foi comparado com aqueles obtidos pelas abordagens híbridas propostas em Bruglieri et al. (2017), onde os autores utilizaram como ambiente de testes um PC Intel Core i7, 3.20 GHz, 6 GB RAM. Como as máquinas utilizadas são diferentes, a comparação em relação ao tempo, não pode ser feita de forma totalmente justa, tornando a comparação no tempo, somente uma relação.

O conjunto de parâmetros do algoritmo GRASP proposto foi estabelecido a partir de testes empíricos com diferentes cenários, tendo-se chegado à seguinte configuração: critério de parada: 70 iterações sem melhora com limite de 1.000 iterações; tamanho do bloco de atualização do vetor de probabilidades: 50 iterações; no Algoritmo 1, fixou-se  $t$  em 80%,  $\kappa = 0.25$ ,  $\rho = 0.5$  e  $\phi = 0.25$ ;  $\alpha$ , usado como parâmetro para determinar a porcentagem em que será realizado o sorteio das inserções, tem valores entre 0.3, a 0.9. Após uma análise inicial das características das instâncias e dos custos das soluções, para função guia (Equação 3.1) adotou-se  $M = 1000$ .

As Tabelas 4.1, 4.2 e 4.3 apresentam a comparação entre o GRASP proposto e os algoritmos VNSB e TPM de Bruglieri et al. (2017). Para cada algoritmo, são mostrados

o *Custo* (somatório do tempo de deslocamento, tempo de atendimento, tempo de espera e tempo de recarga);  $nV$ , número de veículos utilizados; e  $T(s)$ , tempo de processamento em segundos. Os valores em negrito indicam os melhores resultados e a coluna *Desvio* indica o *gap* entre o algoritmo proposto e a melhor solução da literatura.

Observa-se que, sobre o primeiro objetivo do problema (número de veículos), a abordagem proposta obteve os mesmos resultados da literatura em 35 das 36 instâncias testadas. Conforme apresentado em Bruglieri et al. (2017), para as instâncias com cinco clientes, Tabela 4.1, os valores indicam a solução ótima conhecida. Para este grupo de 12 instâncias, o GRASP obteve soluções com custos iguais aos obtidos pelos algoritmos da literatura em nove delas e, onde não alcançou estes resultados, não houve diferença significativa nos custos das soluções. O tempo médio da abordagem proposta ficou em torno de 61 vezes inferior ao algoritmo de três fases TPM, que apresentou os melhores resultados em Bruglieri et al. (2017).

Para as instâncias com 10 clientes, Tabela 4.2, o GRASP não alcançou os melhores resultados em somente três das 12 instâncias, porém com um *gap* médio de apenas 0,20% e com tempo quase 16 vezes inferior ao algoritmo mais eficiente.

Para o grupo de 12 instâncias com 15 clientes, é possível observar na Tabela 4.3 que a abordagem proposta obteve soluções melhores em sete delas e o *gap* médio neste conjunto de instâncias foi  $-5.72\%$ , indicando melhora em relação à literatura, e com tempo mais que 14 vezes inferior ao tempo obtido no trabalho de referência, onde os autores utilizam um hardware superior ao utilizado nos experimentos realizados.

Vale ressaltar que, para a instância R202-15, o custo obtido é maior do que o apresentado na literatura, mas como o número de veículos utilizados na solução é menor e este critério é hierarquicamente superior ao tempo total das rotas, a solução obtida é melhor que a apresentada na literatura. A Figura 4.1 mostra a rota obtida para esta instância. Destaca-se ainda a solução obtida para a instância RC202-15, Figura 4.2, onde o algoritmo proposto alcançou solução com custo 48,39% inferior ao melhor dos dois algoritmos da literatura.

Tabela 4.1: Resultados para instâncias com 5 clientes

Instância	VNSB			TPM			GRASP-RVND			
	Custo	nV	T(s)	Custo	nV	T(s)	Custo	nV	Desvio	T(s)
C101-5	<b>1.262,84</b>	2	0,70	<b>1.262,84</b>	2	0,81	<b>1.262,84</b>	2	0,00%	0,13
C103-5	<b>987,87</b>	1	0,29	<b>987,87</b>	1	0,28	<b>987,87</b>	1	0,00%	0,21
C206-5	<b>1.296,82</b>	1	14,16	<b>1.296,82</b>	1	0,56	1.296,85	1	0,00%	0,27
C208-5	<b>984,80</b>	1	1,31	<b>984,80</b>	1	1,42	984,82	1	0,00%	0,22
R104-5	<b>196,17</b>	2	2,43	<b>196,17</b>	2	0,07	<b>196,17</b>	2	0,00%	0,10
R105-5	<b>231,59</b>	2	0,33	<b>231,59</b>	2	0,09	<b>231,59</b>	2	0,00%	0,06
R202-5	<b>234,16</b>	1	1,62	<b>234,16</b>	1	0,34	234,17	1	0,00%	0,41
R203-5	<b>287,09</b>	1	1,81	<b>287,09</b>	1	0,31	<b>287,09</b>	1	0,00%	0,55
RC105-5	<b>314,31</b>	2	94,01	<b>314,31</b>	2	215,55	<b>314,31</b>	2	0,00%	0,09
RC108-5	<b>342,32</b>	2	1,75	<b>342,32</b>	2	0,39	<b>342,32</b>	2	0,00%	0,15
RC204-5	<b>264,86</b>	1	49,16	<b>264,86</b>	1	0,44	<b>264,86</b>	1	0,00%	0,55
RC208-5	<b>253,17</b>	1	8,00	<b>253,17</b>	1	0,42	<b>253,17</b>	1	0,00%	0,86
<b>MÉDIAS</b>			<b>14,63</b>			<b>18,39</b>			<b>0,00%</b>	<b>0,30</b>

Tabela 4.2: Resultados para instâncias com 10 clientes

Instância	VNSB			TPM			GRASP-RVND			
	Custo	nV	T(s)	Custo	nV	T(s)	Custo	nV	Desvio	T(s)
C101-10	<b>2.335,20</b>	3	183,43	<b>2.335,20</b>	3	545,76	<b>2.335,20</b>	3	0,00%	1,17
C104-10	1.611,72	2	73,43	1.584,90	2	252,95	<b>1.584,89</b>	2	0,00%	3,91
C202-10	2.949,77	1	5,62	2.949,77	1	3,83	<b>2.949,74</b>	1	0,00%	3,21
C205-10	<b>2.525,77</b>	2	0,79	2.539,88	2	1,45	2.539,88	2	0,56%	2,62
R102-10	<b>443,62</b>	3	33,96	<b>443,62</b>	3	0,35	<b>443,62</b>	3	0,00%	1,00
R103-10	<b>347,70</b>	2	93,49	<b>347,70</b>	2	12,19	<b>347,70</b>	2	0,00%	1,83
R201-10	536,38	1	21,70	<b>536,05</b>	1	76,03	545,69	1	1,80%	5,98
R203-10	<b>527,68</b>	1	4,62	<b>527,68</b>	1	9,1	<b>527,68</b>	1	0,00%	10,53
RC102-10	<b>571,26</b>	4	59,01	<b>571,26</b>	4	24,76	<b>571,26</b>	4	0,00%	0,67
RC108-10	<b>493,23</b>	3	143,71	<b>493,23</b>	3	12,73	<b>493,23</b>	3	0,00%	1,55
RC201-10	793,52	1	40,46	<b>793,33</b>	1	9,25	793,52	1	0,02%	5,02
RC205-10	<b>611,66</b>	2	37,51	<b>611,66</b>	2	4,70	<b>611,66</b>	2	0,00%	6,95
<b>MÉDIAS</b>			<b>58,14</b>			<b>79,43</b>			<b>0,20%</b>	<b>3,70</b>

Tabela 4.3: Resultados para instâncias com 15 clientes

Instância	VNSB			TPM			GRASP-RVND			
	Custo	nV	T(s)	Custo	nV	T(s)	Custo	nV	Desvio	T(s)
C103-15	5.254,93	3	240,46	2.446,99	3	178,86	<b>2.352,85</b>	3	-3,85%	14,80
C106-15	2.173,09	3	106,30	2.173,09	3	213,12	<b>2.088,63</b>	3	-3,89%	4,27
C202-15	3.664,12	2	139,83	<b>3.599,80</b>	2	61,15	<b>3.599,80</b>	2	0,00%	25,06
C208-15	<b>2.819,47</b>	2	591,57	<b>2.819,47</b>	2	72,26	<b>2.819,47</b>	2	0,00%	22,97
R102-15	1.241,80	5	559,52	<b>656,37</b>	5	113,75	656,39	5	0,00%	3,53
R105-15	567,82	4	487,02	567,82	4	2,23	<b>567,81</b>	4	0,00%	4,45
R202-15	839,74	2	349,29	840,99	2	58,96	956,22	<b>1</b>	*	68,14
R209-15	621,79	1	330,38	640,64	1	5.718,31	<b>611,23</b>	1	-1,70%	69,15
RC103-15	1.074,90	4	251,55	614,13	4	31,81	<b>614,12</b>	4	0,00%	6,12
RC108-15	974,88	3	756,61	596,17	3	288,01	<b>573,68</b>	3	-3,77%	7,74
RC202-15	3.222,49	2	444,54	1.611,29	2	71,87	<b>831,53</b>	2	-48,39%	36,51
RC204-15	685,61	1	617,22	685,86	1	4.119,53	<b>676,76</b>	1	-1,29%	79,82
<b>MÉDIAS</b>			<b>406,19</b>			<b>910,82</b>			<b>-5,72%</b>	<b>28,55</b>

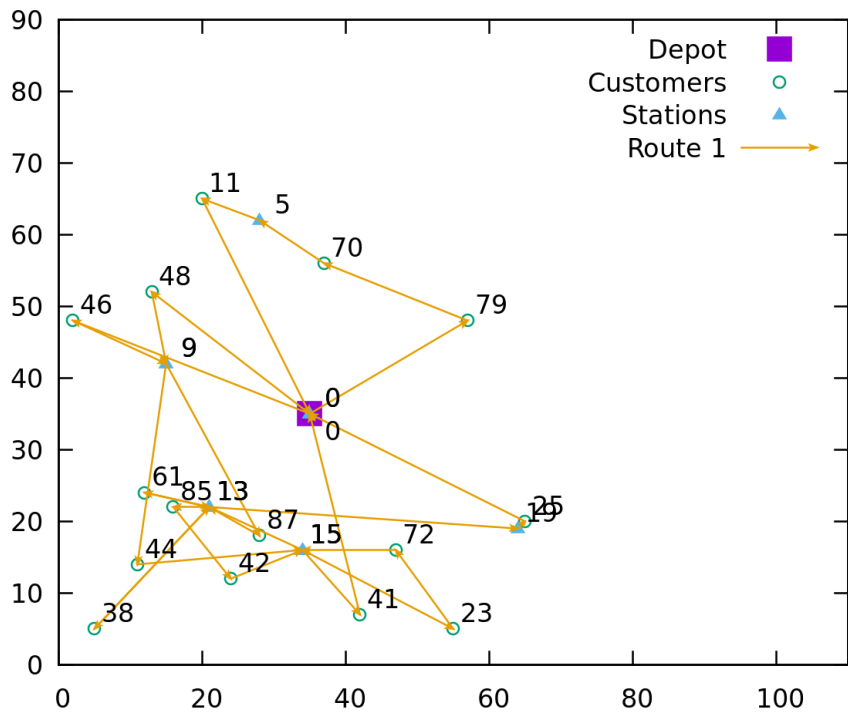


Figura 4.1: Solução para a instância R202C15

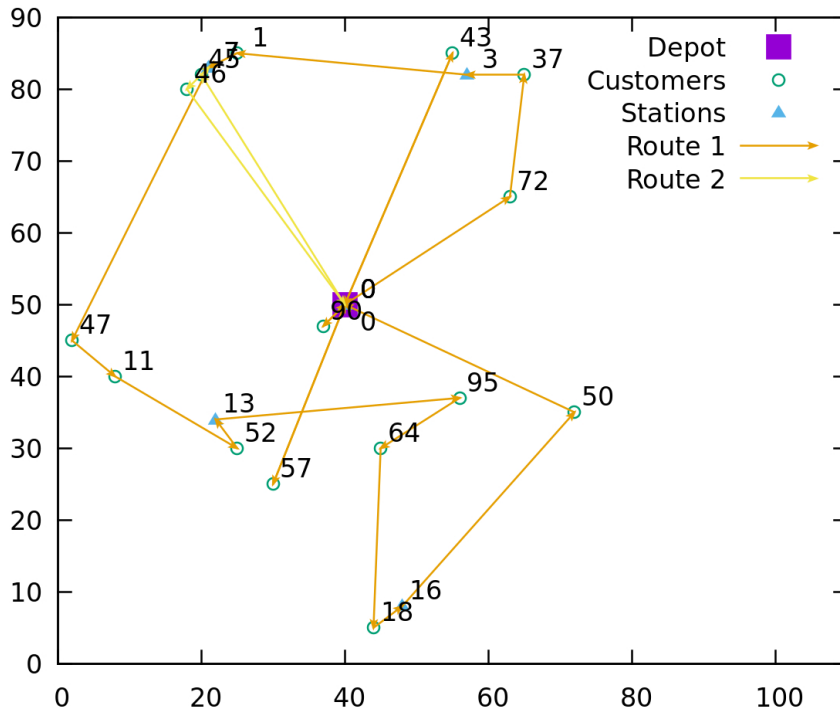


Figura 4.2: Exemplo de Solução Para a Instância RC202C15

A fim de testar a abordagem para instâncias maiores e ainda qualificar os resultados, foram realizados experimentos com algumas instâncias com 100 clientes utilizadas para avaliar o problema EVRPTW. Antes de analisar os resultados apresentados na Tabela 4.4, comparando os resultados com aqueles disponíveis em Schneider, Stenger e Goeke (2014), algumas considerações devem ser feitas:

- Na coluna literatura, temos o resultado da melhor abordagem proposta em Schneider, Stenger e Goeke (2014) para cada instância.
- Todas as instâncias possuem 100 clientes.
- A abordagem proposta em Schneider, Stenger e Goeke (2014) lida com o EVRPTW, onde o segundo objetivo é minimizar a distância total percorrida nas rotas, enquanto no EVRPTW-PR, deseja-se minimizar o tempo total gasto nas rotas, portanto, o custo exibido na Tabela 4 refere-se a distância percorrida pelos veículos nas rotas em ambas abordagens, o que faz com que a comparação direta em relação ao custo não possa ser feita de forma justa, pois os algoritmos seguem objetivos diferentes, mas ainda sim, pode ser comparado o número de veículos obtido, já que os dois problemas tem a minimização de número de veículos como objetivo principal.

- O EVRPTW considera somente recargas totais ao longo do trajeto, o que é uma diferença que na teoria não deveria causar muito impacto, visto que no EVRPTW-PR considera-se uma recarga parcial sempre que seja necessário ou vantajoso, podendo-se optar por efetuar uma recarga total a qualquer momento.
- Os resultados disponibilizados em Schneider, Stenger e Goeke (2014) não trazem o tempo gasto para cada instância, portanto essa coluna não consta na tabela.

Para essa avaliação, alguns parâmetros foram ajustados, como os valores de  $\alpha$  variando entre 0.05 e 0.5,  $t = 15\%$ , sendo somente 70 iterações e 1 execução para cada instância, visto que o tempo de execução estava acima do esperado, e algumas tentativas de diminuí-lo não foram bem sucedidas até o momento.

Tabela 4.4: Tabela de comparação das instâncias de 100 clientes

Instância	Literatura		GRASP-RVND		
	Custo	nV	Custo	nV	T(s)
c101_21	1.053,83	12	1.621,92	14	1.375,82
r201_21	1.264,82	3	1.725,50	4	13.417,57
rc101_21	1.731,07	16	1.993,59	19	2.283,74
rc201_21	1.444,94	4	2.215,63	5	12.366,73

## 5 Conclusões e Trabalhos Futuros

Este trabalho traz uma proposta de algoritmo GRASP Reativo para o problema de roteamento elétrico com janelas de tempo e recargas parciais (E-VRPTW-PR). Na fase de construção, utiliza-se o conceito de VIP para indicar a prioridade de inclusão dos clientes nas rotas e, na fase de busca local, foi utilizado um algoritmo do tipo RVND, onde são empregadas três estruturas de vizinhança e dois procedimentos de melhoria em termos de viabilização e de otimização de rotas. Testes computacionais mostraram que, com tempo de processamento muito inferior às abordagens da literatura, foi possível obter soluções com custos melhores ou iguais em 33 das 36 instâncias testadas, sendo que em sete delas o algoritmo proposto obteve soluções estritamente melhores.

já nas instâncias com 100 clientes (tabela 4.4), os resultados não são satisfatórios, principalmente em relação ao tempo de execução, mas já indicam onde se deve dar o próximo passo para a melhoria do algoritmo, que passa a ser estudar melhor a questão da escalabilidade, que corresponde a analisar o impacto das estruturas utilizadas e identificar quais funções consomem uma maior quantidade de tempo quando o tamanho das instâncias cresce, a fim de fazer um melhor uso da máquina para que seja viável executar testes com um maior número de interações, gerando assim resultados melhores, tanto na questão do custo, quanto na quantidade de veículos.

Como trabalhos futuros, uma vez que o problema traz dois objetivos com hierarquia bem definida, propõe-se o estudo e desenvolvimento de estratégias de penalização destes dois objetivos ao longo da exploração do espaço de busca, assim como estudar os casos em que o código não se comporta como desejado, para que possa ser otimizado de maneira geral, com o intuito de que seja capaz de obter soluções competitivas também para as instâncias de 100 clientes.

## Bibliografia

- BRUGLIERI, M. et al. A three-phase matheuristic for the time-effective electric vehicle routing problem with partial recharges. In: *ScienceDirect*. [S.l.]: Elsevier, 2017. v. 58, p. 95–102.
- BRUGLIERI, M. et al. A matheuristic for the electric vehicle routing problem with time windows. 05 2015.
- BRUGLIERI, M. et al. A variable neighborhood search branching for the electric vehicle routing problem with time windows. *Electronic Notes in Discrete Mathematics*, v. 47, p. 221 – 228, 2015. ISSN 1571-0653. The 3rd International Conference on Variable Neighborhood Search (VNS'14). Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1571065314000717>>.
- DEKKER, R.; BLOEMHOF, J.; MALLIDIS, I. Operations research for green logistics: An overview of aspects, issues, contributions and challenges. In: . [S.l.: s.n.], 2012.
- ERDOĞAN, S.; MILLER-HOOKS, E. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, v. 48, n. 1, p. 100–114, 2012. Disponível em: <<https://EconPapers.repec.org/RePEc:eee:tran:v:48:y:2012:i:1:p:100-114>>.
- FELIPE Ángel et al. A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review*, v. 71, p. 111 – 128, 2014. ISSN 1366-5545. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1366554514001574>>.
- FEO, T. A.; RESENDE, M. G. *Greedy Randomized Adaptive Search Procedures*. 6. ed. [S.l.]: Springer, 1995. 109–134 p.
- FISCHETTI, M.; LODI, A. Local branching. *Math. Program.*, Springer-Verlag, Berlin, Heidelberg, v. 98, n. 1-3, p. 23–47, set. 2003. ISSN 0025-5610. Disponível em: <<https://doi.org/10.1007/s10107-003-0395-5>>.
- HANSEN, P.; MLADENVIĆ, N. An introduction to variable neighborhood search. *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, p. 433–458, 1999.
- HIERMANN, G. et al. The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, v. 252, n. 3, p. 995–1018, 2016.
- HU, B.; RAIDL, G. R. Variable neighborhood descent with selfadaptive neighborhood-ordering. In: *Proceedings of the 7th EU/MEeting on Adaptive, Self-Adaptive, and Multi-Level Metaheuristics*. [S.l.: s.n.], 2006.
- KESKIN, M.; ÇATAY, B. Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, v. 65, p. 111–127, 2016.



PRAIS, M.; RIBEIRO, C. C. Reactive grasp: An application to a matrix decomposition problem in tdma traffic assignment. *INFORMS Journal on Computing*, v. 12, n. 3, p. 164–176, 2000.

RESENDE, M. G.; RIBEIRO, C. C. Greedy randomized adaptive search procedures: Advances, hybridizations, and applications. In: \_\_\_\_\_. *Handbook of Metaheuristics*. Boston, MA: Springer US, 2010. p. 283–319.

SBIHI, A.; EGGLESE, R. Combinatorial optimization and green logistics. In: . [S.l.: s.n.], 2010.

SCHNEIDER, M.; STENGER, A.; GOEKE, D. The electric vehicle-routing problem with time windows and recharging stations. In: *Transportation Science*. [S.l.]: Informs, 2014. v. 48, p. 500–520.

SHAO, S. et al. Electric vehicle routing problem with charging time and variable travel time. *Mathematical Problems in Engineering*, v. 2017, p. 1–13, 01 2017.

SUBRAMANIAN, A. et al. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers Operations Research*, v. 37, n. 11, p. 1899 – 1911, 2010. ISSN 0305-0548.

ĆIROVIĆ, G.; PAMUČAR, D.; BOŽANIĆ, D. Green logistic vehicle routing problem: Routing light delivery vehicles in urban areas using a neuro-fuzzy model. *Expert Systems with Applications*, v. 41, n. 9, p. 4245 – 4258, 2014. ISSN 0957-4174. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0957417414000177>>.