

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

**Abordagem híbrida para o problema de  
roteamento de veículos elétricos com  
localização de estações de troca de baterias**

**Bráulio Manoel de Oliveira Portela**

JUIZ DE FORA  
JUNHO, 2018

# Abordagem híbrida para o problema de roteamento de veículos elétricos com localização de estações de troca de baterias

BRÁULIO MANOEL DE OLIVEIRA PORTELA

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: Stênio Sã Rosário Furtado Soares

JUIZ DE FORA  
JUNHO, 2018

ABORDAGEM HÍBRIDA PARA O PROBLEMA DE  
ROTEAMENTO DE VEÍCULOS ELÉTRICOS COM LOCALIZAÇÃO  
DE ESTAÇÕES DE TROCA DE BATERIAS

Bráulio Manoel de Oliveira Portela

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

Aprovada por:

Luciana Brugiolo Gonçalves  
DSc em Computação - UFF

Stênio Sã Rosário Furtado Soares  
DSc em Computação - UFF

Lorenza Leão Oliveira Moreno  
DSc em Informática - PUC RJ

Heder Soares Bernardino  
DSc em Modelagem Computacional - LNCC

JUIZ DE FORA  
26 DE JUNHO, 2018

*Aos meus familiares e amigos.*

## Resumo

Os impactos ambientais vem sendo um assunto cada vez mais pertinente. Uma preocupação atual é o grande impacto causado por emissão de poluentes vinda de automóveis movidos a combustão. Neste trabalho é estudada uma forma de utilizar veículos elétricos em aplicações de logística. Para isso é estudado o problema de roteamento de veículos elétricos e de localização de estações de troca de baterias. Este busca determinar a localização de estações de troca de bateria e simultaneamente o roteamento de uma frota de veículos elétricos. O intuito deste problema é amenizar os impactos ambientais causados pelo uso de automóveis e possibilitar a utilização de veículos elétricos de forma economicamente viável em serviços de transportes. Para tal fim é apresentada uma nova perspectiva e uma abordagem híbrida combinando heurística com Programação Matemática. Esta técnica é realizada através de um modelo, aplicado à um resolvedor, para solucionar este problema. Os experimentos foram realizados a partir de dados gerados pela literatura e também por modificações propostas neste trabalho. Os resultados obtidos neste estudo mostram a contribuição que o método híbrido traz aos resultados, sem um impacto tão grande ao tempo de execução, provando-se ser uma abordagem competitiva.

**Palavras-chave:** Veículos Elétricos, Problema de roteamento e localização, Troca de Bateria, Abordagem heurística, Abordagem híbrida, Logística Verde.

# Abstract

Environmental impacts have become an increasingly relevant issue. A current concern is the great impact caused by the emission of pollutants from combustion-fueled automobiles. In this work we study a way to use electric vehicles in logistics applications. For this, the problem of routing of electric vehicles and the location of battery exchange stations is studied. This seeks to determine the location of battery exchange stations and simultaneously the routing of a fleet of electric vehicles. The purpose of this problem is to mitigate the environmental impacts caused by the use of automobiles and to enable the use of electric vehicles in an economically viable way in transport services. For this purpose a new perspective and a hybrid approach are presented combining heuristics with Mathematical Programming. This technique is performed through a model, applied to a resolver, to solve this problem. The experiments were carried out from data generated by the literature and also by modifications proposed in this work. The results obtained in this study show the contribution that the hybrid method brings to the results, without a great impact to the execution time, proving to be a competitive approach.

**Keywords:** Electric Vehicles, Routing and localization problem, Battery change, Heuristic approach, Hybrid approach, Green Logistics.

## Agradecimentos

Aos meus pais Elen e Wilmar e minha irmã Thalita por todo amor, força, encorajamento e assistência.

Aos demais familiares pela confiança e incentivo.

Aos meus amigos por todo apoio e carinho.

À amiga Thais e família por todo carinho e suporte durante os anos da graduação.

Aos amigos do CGCO/UFJF por toda experiência e ensinamentos que me proporcionaram.

Aos amigos do CEAD/UFJF pela oportunidade profissional, apoio e confiança.

Aos professores Luciana 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.

*“Lost in thought and lost in time, while  
the seeds of life and the seeds of change  
were planted...”*

*I knew the moment had arrived for kil-  
ling the past and coming back to life.”*

*Pink Floyd (Coming Back To Life)*

# Conteúdo

<b>Lista de Figuras</b>	<b>7</b>
<b>Lista de Tabelas</b>	<b>8</b>
<b>Lista de Abreviações</b>	<b>9</b>
<b>1 Introdução</b>	<b>10</b>
1.1 O problema estudado . . . . .	11
1.2 Justificativa . . . . .	12
1.3 Objetivos . . . . .	12
<b>2 Revisão bibliográfica</b>	<b>13</b>
2.1 Problemas clássicos . . . . .	13
2.2 Problema de localização e roteamento . . . . .	14
2.3 BSS-EV-LRP . . . . .	15
2.3.1 Modelo . . . . .	16
<b>3 Abordagem proposta</b>	<b>20</b>
3.1 Abordagem Heurística . . . . .	21
3.1.1 <i>Distribuição dos clientes em rotas</i> . . . . .	21
3.1.2 <i>Refinamento das rotas</i> . . . . .	23
3.1.3 <i>Designação das estações de troca de bateria</i> . . . . .	25
3.1.4 <i>Multi-start</i> . . . . .	29
3.2 Abordagem Híbrida . . . . .	30
3.2.1 <i>Modelo por rotas</i> . . . . .	30
3.2.2 <i>Geração de conjunto de rotas</i> . . . . .	31
3.2.3 <i>Refinamento via Programação Matemática</i> . . . . .	32
<b>4 Resultados computacionais</b>	<b>35</b>
4.1 Instâncias e parâmetros . . . . .	35
4.2 Resultados . . . . .	36
<b>5 Conclusões e trabalhos futuros</b>	<b>43</b>
<b>Bibliografia</b>	<b>45</b>

## Lista de Figuras

3.1	Exemplo da representação de uma solução em um vetor de vetores. . . . .	20
3.2	Demonstração da busca local <i>Cross</i> utilizada pelo RVND. . . . .	23
3.3	Demonstração do algoritmo SDM. . . . .	28
3.4	Exemplo de derivação de rotas . . . . .	32

## Lista de Tabelas

3.1	Lista de Buscas Locais utilizadas pelo RVND. . . . .	23
4.1	Comparativo dos resultados entre CPLEX, IMB-SDM-RVND e IMB-SDM-RVND-PL para as instâncias de tamanho muito pequenas. . . . .	37
4.2	Comparativo dos resultados entre IMB-SDM-RVND e IMB-SDM-RVND-PL para as instâncias de pequeno e médio porte. . . . .	38
4.3	Comparativo dos resultados entre a abordagem híbrida IMB-SDM-RVND-PL e a abordagem AVNS. . . . .	40
4.4	Comparativo dos resultados entre a abordagem híbrida IMB-SDM-RVND-PL e a abordagem SIGALNS. . . . .	42

## Lista de Abreviações

DCC	Departamento de Ciência da Computação
UFJF	Universidade Federal de Juiz de Fora
VE	Veículo elétrico
ETB	Estação de troca de baterias
PRV	Problema de Roteamento de Veículos
PCV	Problema do caixeiro viajante
PLR	Problema de localização e roteamento
IMB	Inserção Mais Barata
SDM	Subconjunto Dominante Mínimo
IETB	Inserção de de troca de baterias

# 1 Introdução

Nos dias atuais, os aspectos ambientais assumem cada vez mais importância. A busca por novas soluções para reduzir os impactos ambientais se tornou uma preocupação de vários países e representa o dia a dia de várias empresas e pesquisadores pelo mundo. Esta nova cultura levou ao surgimento de uma área de pesquisa chamada Logística Verde, que visa buscar e viabilizar alternativas com menor agressão ao meio ambiente nas ações da área de logística. Um de seus principais assuntos é a utilização de Veículos Elétricos como alternativa mais sustentável em problemas de roteamento.

Sabe-se que os automóveis foram importantíssimos para o desenvolvimento da humanidade nos últimos séculos, sendo utilizados como transportes de pessoas ou cargas por curtas e longas distâncias. Os veículos predominantemente em circulação hoje utilizam motor a combustão interna, alimentados por combustíveis fósseis que, apesar de alguns pontos positivos, como eficiência e autonomia, possuem sérias desvantagens. A massiva poluição do ar, principalmente em áreas urbanas, e a crescente preocupação com o aquecimento global são motivações suficientes para o desenvolvimento de alternativas mais sustentáveis.

Como apresentado em Baran e Legey (2011), projetos de veículos elétricos já existem desde a virada do século XX, mas apenas recentemente, a partir de incentivos governamentais e o desenvolvimento de novas tecnologias de bateria, tornou-se uma alternativa viável, como avaliado em Castro e Ferreira (2010). Entretanto, a autonomia dos veículos elétricos continua sendo relativamente menor quando comparada a dos transportes com motores a combustão.

Um dos cenários com maiores dificuldades para a substituição de veículos a combustão por elétricos é a cadeia de suprimentos, também conhecida como rede logística, por demandar viagens mais longas e atendimento a vários pontos consecutivos sem o retorno ao ponto de partida ou pontos de abastecimento.

Com uma autonomia menor dos carros elétricos e com a necessidade de longas viagens, cria-se uma certa barreira para que o modelo mais adotado atualmente seja

alterado para a utilização de veículos elétricos. Com isso, métodos para se economizar energia são importantíssimos, e um deles é a utilização de modelos matemáticos que determinem os melhores caminhos, os mais curtos, ou os que consomem menos do motor para alcançar os locais desejados. Além destes aspectos, um importante fator é considerar a possibilidade de instalação de pontos de recarga ou de troca de bateria ao longo das rotas, caso seja necessário.

Assim, a instalação de pontos de troca de bateria apresenta-se como uma boa alternativa para problemas de alcance de longas distâncias em curto tempo com a utilização de veículos elétricos, como é explorado em Mirchandani, Adler e Madsen (2014). Ao invés da necessidade de um gasto elevado de tempo para se recarregar a bateria de um veículo, a troca de sua bateria por uma outra, previamente carregada, em pontos estratégicos demandaria poucos minutos.

## 1.1 O problema estudado

O Problema de Roteamento de Veículos (PRV) é um dos mais estudados na área de otimização combinatória devido a sua complexidade e relevância econômica. Porém, sabe-se que abordagens clássicas como as usadas para o Problema do Caixeiro Viajante (PCV), por exemplo, adaptadas e inseridas neste panorama em específico, baseando-se nas características e limitações da utilização de veículos elétricos, necessitariam de outras abordagens, diferentes daquelas já conhecidas que possam melhor solucioná-las.

Por conta desta limitação, neste trabalho considera-se uma variação do Problema de Roteamento de Veículos baseada na proposta de Yang e Sun (2015), denominada Problema de Roteamento de Veículos Elétricos e de Localização de Estações de Troca de Baterias (BSS-EV-LRP), que foca em abordagens que possam viabilizar a utilização de veículos elétricos no lugar de veículos a combustão tentando reduzir a quantidade de energia consumida e conseqüentemente os impactos ambientais causados no cenário de Cadeia de Suprimentos.

O BSS-EV-LRP é um problema NP-Difícil (HOCHBAUM, 1997) e sua complexidade pode ser observada sobre dois diferentes aspectos do problema: designar a localização ótima de estações de troca de bateria que servirão de suporte para os veículos elétricos

utilizados ao mesmo tempo que determina o roteamento desta frota.

## 1.2 Justificativa

O crescente apelo em relação a preservação do meio ambiente traz para a computação uma nova gama de problemas e desafios nos processos de tomada de decisão.

Por se tratar de um tema relativamente novo, muito ainda há a ser estudado e desenvolvido. Por outro lado, um dos pontos que vem fortalecendo esta linha são as novas legislações, exigências governamentais e apelos da população, que estão levando cada vez mais os órgãos públicos e privados a buscar e incentivar soluções sustentáveis para suas atividades em logística.

O problema tratado neste trabalho, por suas características (novos pontos a serem levados em consideração e sua complexidade), se torna um desafio completamente novo a ser resolvido. Além disto, por se tratar de uma composição de duas diferentes questões a serem solucionadas ao mesmo tempo, o desafio computacional se torna ainda maior.

## 1.3 Objetivos

A proposta deste trabalho é buscar novas abordagens que possam resolver o Problema de Roteamento de Veículos Elétricos e de Localização de Estações de Troca de Baterias, de forma competitiva com o intuito de, além da economia, amenizar os impactos ambientais causados pelo uso de automóveis.

## 2 Revisão bibliográfica

Como citado anteriormente, o problema que este trabalho se propõe a estudar pode ser dividido em duas questões tratadas em paralelo. A primeira, uma definição de rotas e a segunda, uma designação de locais de facilitadores, formando assim a junção de dois problemas já conhecidos, o problema de roteamento de veículos, descrito de forma abrangente no livro *The Vehicle Routing Problem* (TOTH; VIGO, 2014), e o problema de localização descrito em Lorena (2001).

### 2.1 Problemas clássicos

Roteamento de veículos é um tema fortemente estudado pela comunidade científica há varias décadas, sendo considerado uns dos problemas clássicos na área de otimização. Seu objetivo consiste em determinar, dentre todas as possíveis rotas disponíveis, qual a que representa o menor custo. A solução para este tipo de problema, a primeira vista, pode ser considerada simples de se alcançar, mas facilmente ela pode se tornar muito complexa quando a quantidade de nós (destinos) e conseqüentemente suas combinações disponíveis de caminhos fica muito grande.

O problema de roteamento de veículos (PRV), proposto em 1959 por Dantzig e Ramser (DANTZIG; RAMSER, 1959), é uma generalização do também clássico problema do caixeiro viajante (PCV), pertencentes a classe de problemas NP-Difíceis, ou seja, são problemas para os quais não se conhece algoritmos polinomiais determinísticos que os resolvam (TOTH; VIGO, 2002). O PRV pode ser exemplificado como, dado um depósito de abastecimento e vários clientes ligados por um ou vários caminhos ao depósito e entre si com custos (ou distâncias) diferentes, determinar as melhores rotas a serem usadas por um ou mais veículos, que saiam do depósito, passem por todos os clientes e retorne ao ponto de partida com o menor custo possível (CLARKE; WRIGHT, 1964).

O problema de localização combinado ao de roteamento, em grande parte, consiste em determinar, dentre locais candidatos, aqueles que melhor sirvam como suporte para o

problema. Estes locais, dependendo do cenário, podem ser qualquer tipo de facilitador, como depósitos de mercadorias, uma sede da empresa, um hotel (LORENA, 2001) ou, no caso tratado por este trabalho, um ponto de trocas de bateria.

## 2.2 Problema de localização e roteamento

O problema de localização e roteamento (PLR) integra estes dois níveis de decisão com o objetivo de solucioná-los simultaneamente. Há, nas últimas décadas, muitas variantes da PLR sendo estudadas, como: PLR com múltiplos depósitos (LAPORTE; NOBERT; TAILLEFER, 1988; WU; LOW; BAI, 2002), PLR Capacitados (capacidades em depósitos ou em veículos) (LYSGAARD; LETCHFORD; EGGLESE, 2004; LIN et al., 2009), PLR com janela de tempo (ZARANDI; HEMMATI; DAVARI, 2011; ZARANDI et al., 2013), entre outros.

Há algum tempo, alguns estudos foram publicados sobre questões do PLR (MIN; JAYARAMAN; SRIVASTAVA, 1998; NAGY; SALHI, 2007) compilando vários trabalhos e questões relacionadas. Alguns algoritmos foram propostos para melhor resolver o problema, como algoritmo branch and bound no trabalho desenvolvido por Laport e Norbert (LAPORTE; NOBERT, 1981), algoritmo usado principalmente para o problema do caixeiro viajante e o problema da mochila. Foi também desenvolvido por Laporte, Nobert e Arpin (1986) o algoritmo branch and cut para PLR não-capacitados, que obteve resultados satisfatórios para pequenas e médias instâncias.

Há alguns estudos voltados para a designação de pontos de abastecimento de veículos de combustível, assunto que nas ultimas décadas ganhou bastante atenção. Com base no Modelo de Localização de Captura de Fluxo (FCLM) proposto por Hodgson e por Berman et al. (HODGSON, 1990; BERMAN; LARSON; FOUSKA, 1992), Kuby e Lim estudaram o modelo de locação de fluxo de reabastecimento (FRLM) que localiza um número determinado de estações para maximizar o fluxo de veículos sobre os caminhos com a consideração do alcance destes transportes (KUBY; LIM, 2007).

## 2.3 BSS-EV-LRP

Recentemente, buscando uma redução do impacto ambiental causado pelas atividades relacionadas à produção e distribuição de produtos, foi criada uma linha de pesquisa chamada Logística Verde que visa buscar e viabilizar alternativas com menor agressão ao meio ambiente, descrita de forma geral em Sbihi e Eglese (2007), Sbihi e Eglese (2010). Um de seus principais assuntos é a utilização de Veículos Elétricos como uma melhor alternativa em práticas logísticas.

Como é exposto em Mirchandani, Adler e Madsen (2014), há vários motivos para se acreditar na substituição das frotas de muitas empresas de transporte por veículos com energia mais sustentável, como os veículos elétricos. E uma de suas soluções são as designações de estações de troca de bateria para servirem como suporte para veículos elétricos atenderem à malha de clientes baseados em suas baterias limitadas e consequentemente sua autonomia reduzida.

O Problema de Roteamento de Veículos Elétricos e de Localização de Estações de Troca de Baterias é um problema proposto em Yang e Sun (2015) que visa determinar a localização de estações de troca de bateria (ETBs) e o plano de roteamento de uma frota de veículos elétricos (VEs). As ETBs são determinadas estrategicamente, considerando um custo de implementação, dentre um conjunto de pontos candidatos. O propósito do problema é encontrar a melhor solução buscando minimizar a soma do custo de implementações de ETBs e os gastos de energia dos veículos ao efetuarem suas rotas. No trabalho de Yang e Sun (2015), os autores apresentam modelos para o problema e duas diferentes combinações de algoritmos para resolvê-lo. A primeira combinação é denominada de heurística TS-MCWS que combina Busca Tabu, usada para procurar a estratégia de localização e o algoritmo de Economia de Clarke e Wright Modificado que compromete-se à decisão de roteamento baseado nesta solução de localização. A segunda proposta destes autores é a chamada SIGALNS formada pela combinação de *modified Sweep heuristic*, *Iterated Greedy*, *Adaptive Large Neighborhood Search* e uma heurística de melhoria para o *EV location routing problem*. Além disso, é também apresentada uma profunda análise econômica e ambiental, incluindo a comparação entre cenários básicos e ampliados, mostrando a eficiência e redução de emissões quando utilizado VEs em práticas de

logística.

No trabalho de Hof, Schneider e Goeke (2017), além de descrever mais detalhadamente o problema e principalmente as instâncias utilizadas nos processos de testes (mostrando um passo a passo de como adaptá-las para o problema), esta publicação traz uma nova abordagem, mais eficaz, para solucionar o problema proposto. Utilizando um algoritmo AVNS (*Adaptive Variable Neighborhood Search*), a estratégia se mostrou capaz de reduzir o número de ETBs utilizado nas soluções em relação aos resultados da literatura. Sendo assim, neste trabalho, são usados estes resultados como comparativos.

### 2.3.1 Modelo

A seguir é apresentado uma compilação do modelo matemático apresentado em Yang e Sun (2015) para o problema.

Conjuntos:

- $I$  conjunto de clientes;
- $J$  conjunto de candidatos a ETBs;
- $K$  conjunto de VEs;
- $R$  conjunto de rotas de veículos;
- $\{o\}$  o único depósito;
- $\{o'\}$  uma cópia do depósito  $\{o\}$ ;
- $V$  conjunto de nós,  $V = I \cup J \cup \{o\} \cup \{o'\}$ .

Variáveis de decisão:

- $y_j$  variável de decisão binária, tendo um valor de 1 se uma ETB estiver estabelecida no nó  $j$  e 0 caso contrário;
- $x_{ghk}$  variável de decisão binária, tendo um valor de 1 se o veículo  $k$  vai do nó  $g$  para o nó  $h$  e 0 caso contrário.

Constantes de entrada do problema:

- $d_{gh}$  distância do nó  $g$  para o nó  $h$ ;
- $\gamma_{ghk}$  custo de transporte por milha a partir do nó  $g$  para o nó  $h$  com o veículo  $k$ ;
- $M$  um número grande, relacionado ao limite de baterias trocadas em uma ETB;
- $q_i$  demanda no nó  $i \in I \cup J$ , onde  $q_i = 0, \forall i \in J$ ;
- $U_k$  capacidade de carga do veículo  $k$ ;
- $Q$  distância máxima que uma bateria totalmente carregada permite alcançar;
- $c_j$  custo de designação da unidade ETB no nó  $j$ .

Variáveis auxiliares:

- $u_{ghk}$  capacidade de carga restante quando o veículo  $k$  atinge o nó  $g$  depois de deixar o nó  $h$ ;
- $P_{gk}^1$  distância máxima que a carga restante da bateria permite alcançar quando o veículo  $k$  chega ao nó  $g$ ;
- $P_{gk}^2$  distância máxima que a carga restante da bateria permite alcançar quando o veículo  $k$  deixa o nó  $g$ ;

Modelo:

$$\text{Min } f = \sum_{j \in J} c_j y_j + \sum_{g \in V} \sum_{h \in V} \sum_{k \in K} \gamma_{ghk} d_{gh} x_{ghk} \quad (2.1)$$

Sujeito a:

$$\sum_{g \in V \setminus \{o'\}, g \neq h} \sum_{k \in K} x_{ghk} = 1 \quad \forall h \in I \quad (2.2)$$

$$\sum_{g \in V \setminus \{o'\}, g \neq h} \sum_{k \in K} x_{ghk} \leq M y_h \quad \forall h \in J \quad (2.3)$$

$$\sum_{h \in V \setminus \{o\}, h \neq g} x_{ghk} - \sum_{h \in V \setminus \{o'\}, h \neq g} x_{h g k} = 0 \quad \forall g \in V \setminus \{o, o'\}, k \in K \quad (2.4)$$

$$\sum_{h \in V \setminus \{o\}} x_{ohk} - \sum_{h \in V \setminus \{o'\}} x_{ho'k} = 0 \quad \forall k \in K \quad (2.5)$$

$$\sum_{h \in V \setminus \{o\}} x_{ohk} \leq 1 \quad \forall k \in K \quad (2.6)$$

$$\sum_{k \in K} \sum_{h \in V \setminus \{o\}} x_{ohk} \leq |K| \quad (2.7)$$

$$\sum_{i \in V \setminus \{o', j\}} u_{ijk} = \sum_{i \in V \setminus \{o, j\}} u_{jik} \quad \forall j \in J, k \in K \quad (2.8)$$

$$\sum_{j \in V \setminus \{o, i\}} u_{ijk} \leq \sum_{j \in V \setminus \{o', i\}} u_{jik} - q_i \sum_{j \in V \setminus \{o', i\}} x_{jik} + U_k (1 - \sum_{j \in V \setminus \{o', i\}} x_{jik}) \quad \forall i \in I, k \in K \quad (2.9)$$

$$0 \leq u_{ijk} \leq U_k x_{ijk} \quad \forall i \in V \setminus \{o'\}, j \in V \setminus \{o\}, i \neq j, k \in K \quad (2.10)$$

$$P_{hk}^1 \leq P_{gk}^2 - d_{gh} x_{ghk} + Q(1 - x_{ghk}) \quad \forall g \in V \setminus \{o'\}, h \in V \setminus \{o\}, g \neq h, k \in K \quad (2.11)$$

$$P_{ok}^2 = Q \quad \forall k \in K \quad (2.12)$$

$$P_{gk}^2 = Q y_g \quad \forall g \in J \quad (2.13)$$

$$P_{hk}^2 = P_{hk}^1 \quad \forall h \in I \quad (2.14)$$

$$P_{hk}^1 \geq 0 \quad \forall h \in V \quad (2.15)$$

$$y_j, x_{ghk} \in \{0, 1\} \quad \forall j \in J, g \in V \setminus \{o'\}, h \in V \setminus \{o\}, k \in K \quad (2.16)$$

A função objetivo (2.1) minimiza o custo total, incluindo a soma dos custos de implantação das ETBs e a soma dos custos de deslocamentos dos veículos. As restrições (2.2) garantem que cada cliente seja visitado sempre por um único veículo. O conjunto de restrições (2.3) garante que os VEs só podem trocar suas baterias em uma ETB estabelecida, enquanto as restrições (2.4) garantem o balanço do fluxo para cada veículo em cada nó. As restrições (2.5) garantem que cada veículo deve retornar ao depósito e as restrições (2.6) restringem cada veículo a apenas uma viagem quando este sai do depósito. A desigualdade (2.7) estabelece que a quantidade de veículos é limitado para  $|K|$ . A soma da capacidade de carga restante de um VE chegando em uma ETB é igual à soma da capacidade remanescente deste VE deixando um ETB (2.8). As restrições (2.9) garantem o controle da alteração do nível de carga remanescente de um VE para cada nó, enquanto a desigualdade (2.10) garante a limitação do nível de carga remanescente. As restrições (2.11) rastreiam nível da bateria de um veículo elétrico com base na sequência de nó. O nível de energia da bateria quando VEs deixam o depósito é sempre máxima (2.12). O nível de energia da bateria quando VEs deixam uma ETB estabelecida é sempre máxima (2.13). O nível da bateria permanece inalterada enquanto VEs atendem um nó cliente

(2.14). As restrições (2.15) garantem que cada VE tem energia suficiente em sua bateria para visitar os clientes restantes e retornar ao depósito. E, por último, as restrições (2.16) estabelecem a natureza binária das variáveis de decisão.

Pode-se observar que, com as restrições (2.8) e (2.9), torna-se possível, para um VE, passar por uma ETB mais de uma vez, mas é restrito pela desigualdade (2.2) a visita a um cliente apenas uma vez.

### 3 Abordagem proposta

Este trabalho tem como principal objetivo o desenvolvimento de uma abordagem híbrida, combinando métodos matemáticos e heurísticos, usando modelo complementar e combinando-o com heurísticas para melhor resolver o Problema de Roteamento de Veículos Elétricos e de Localização de Estações de Troca de Baterias de forma competitiva.

De maneira simplificada, a metodologia seguida é composta das seguintes etapas:

- Desenvolvimento de uma abordagem heurística, combinando estratégias de construção, inspirado no algoritmo de inserção mais barata e utilização do algoritmo RVND para a definição e o refinamento das rotas e localização de BTBs com auxílio do conceito de subconjunto dominante mínimo.
- Desenvolvimento de abordagem matemática para refinamento das localizações de ETBs.

Para a representação de solução do problema foi utilizado um vetor de rotas onde cada rota é representada por um vetor de pontos. A Figura 3.1 ilustra esta organização onde o ponto 0 representa o ponto inicial e final da rota (depósito) os outros números representam os clientes e, por fim, os pontos marcados com “B” são as ETBs utilizadas nas rotas. Por exemplo, na rota 2 o veículo sai do depósito, visita os clientes 13 e 7, depois vai até a ETB B7, troca sua bateria, e depois visita o cliente 1 e retorna ao depósito.

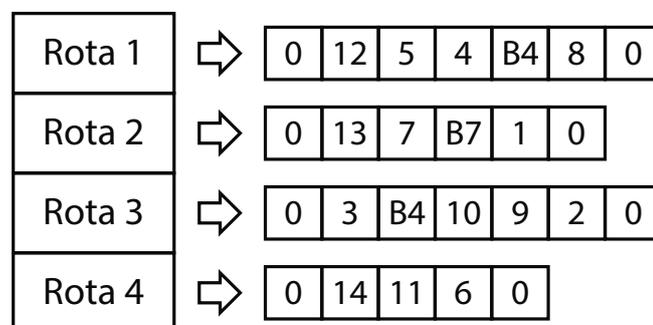


Figura 3.1: Exemplo da representação de uma solução em um vetor de vetores.

## 3.1 Abordagem Heurística

Neste trabalho, inicialmente, é apresentada uma nova abordagem, a IMB-SDM-RVND, introduzida em Portela, Gonçalves e Soares (2017), que consiste em uma combinação de modificações dos Algoritmos de Inserção Mais Barata (ROSENKRANTZ; STEARNS; LEWIS, 1974) para montagem das rotas com um algoritmo que se baseia no conceito do Subconjunto Dominante Mínimo (PANG et al., 2010) para a designação das ETBs e da metaheurística de Busca em Vizinhança Variável Aleatória (RVND) (OSMAN, 1993) para refinamentos das rotas.

A seguir cada um destes módulos são devidamente detalhados.

### 3.1.1 *Distribuição dos clientes em rotas*

Inicialmente, para construir as rotas da solução, é utilizado o algoritmo de Inserção Mais Barata (IMB) com alguns mecanismos de aleatoriedade na escolha e inserção de cada um dos clientes à solução, descrito no Algoritmo 1.

Na Linha 1 do Algoritmo 1 é calculada a soma das demandas  $q_i$  de todos os clientes, dividida pela capacidade  $U$  dos veículos (frota homogênea). Este valor é arredondado para cima e por fim é vinculado à variável  $ve$  que representa a quantidade mínima de veículos necessários para atender as demandas de todos os clientes.

A solução  $S$  recebe, na Linha 2, o resultado da função  $inicializaSol(ve)$  que consiste em criar um vetor de vetores, como na Figura 3.1. É criado um vetor com  $ve$  posições referentes ao número de rotas criadas. Cada uma das rotas recebe os pontos inicial e final que representam o depósito de onde os veículos devem sair no início de um trajeto e retornar ao final dele. Após isso, a lista de clientes  $LC$  recebe o conjunto de todos os clientes  $I$  (Linha 3).

Em seguida, na Linha 4, a solução  $S$  recebe o resultado da função  $addClienteInicial(S)$ , que consiste em uma função que adiciona a cada uma das  $ve$  rotas, um cliente inicial sorteado. Cada cliente sorteado e adicionado a alguma das rotas é retirado do conjunto  $LC$ . O método utilizado para se sortear cada um dos clientes iniciais das rotas é o algoritmo de Roleta (LIPOWSKI; LIPOWSKA, 2012) utilizando a demanda destes clientes como parâmetro onde, quanto maior a demanda de um cliente, mais chances deste

ser escolhido.

Da Linha 5 a 9 é criada uma iteração que continuará se repetindo enquanto a lista de clientes ainda não atendidos,  $LC$ , possui algum elemento. A variável  $i'$ , na Linha 6, recebe o cliente sorteado pela função  $torneioTernario(LC)$  onde se sorteia três clientes e se escolhe aquele que possui o custo mais baixo para ser adicionado a uma das rotas. Este cliente é adicionado pela função  $melhorInsercao(S, i')$ , Linha 7, na posição em que o custo adicional deste novo cliente seja o menor possível para a solução.

Em seguida, na Linha 8, o cliente adicionado é retirado do conjunto  $LC$ . Em qualquer momento deste passo, na função  $melhorInsercao(S, i')$ , caso haja algum cliente que não possa mais ser inserido em nenhuma das rotas existentes por exceder o limite da capacidade de todos os veículos, é criada uma nova rota em  $S$  apenas com os pontos inicial e final e este cliente é imediatamente inserido nesta nova rota. Por fim, a solução parcial é retornada com as rotas criadas. Esta solução é dita parcial por não contemplar as ETBs.

---

**Algoritmo 1: IMB**

---

**Ensure:** Solução parcial  $S$  para roteamento dos VEs;  
 1:  $ve \leftarrow \lceil \sum_{i \in I} q_i / U \rceil$ ;  
 2:  $S \leftarrow inicializaSol(ve)$ ;  
 3:  $LC \leftarrow I$ ;  
 4:  $S \leftarrow addClienteInicial(S, LC)$ ;  
 5: **while**  $|LC| > 0$  **do**  
 6:    $i' \leftarrow torneioTernario(LC)$ ;  
 7:    $S \leftarrow melhorInsercao(S, i')$ ;  
 8:    $LC \leftarrow LC - \{i'\}$ ;  
 9: **end while**  
 10: **return**  $S$ ;

---

Como forma de auxílio aos cálculos de inserção, neste algoritmo foi utilizada uma matriz de melhores inserções que consta com o custo e a posição da melhor inserção de cada cliente em cada uma das rotas previamente calculados. Os dados desta matriz são atualizados de forma otimizada a cada inserção de um novo cliente.

### 3.1.2 Refinamento das rotas

Após as rotas serem geradas pelo Algoritmo 1, é aplicado o método RVND a esta solução parcial. Este método consiste em uma lista de buscas locais com ordem sorteada em que a solução é submetida a procedimentos de busca de refinamentos e melhorias ao resultado. Para isto foi escolhido um conjunto de cinco buscas locais (SUBRAMANIAN, 2012) descritas pela Tabela 3.1. As buscas inter-rotas atuam entre mais de uma rota enquanto as intra-rotas operam em cada rota individualmente.

#	Buscas	Alcance
1	<i>Cross</i>	Inter-rotas
2	<i>Shift(1,0)</i>	Inter-rotas
3	<i>Swap(1,1)</i>	Inter-rotas
4	<i>2-opt</i>	Intra-rotas
5	<i>Reinsertion</i>	Intra-rotas

Tabela 3.1: Lista de Buscas Locais utilizadas pelo RVND.

A busca *Cross* (TAILLARD et al., 1997) se resume ao cruzamento entre duas rotas. Ela busca a combinação de duas arestas, de rotas distintas, que caso sejam desconectadas e reconectadas de outra forma, consigam ter o maior ganho para a solução. Além de quais arestas são escolhidas para serem desconectadas, é escolhida também a melhor reconexão a ser feita, como é possível observar na Figura 3.2. Toda movimentação é feita observando a viabilidade de carga de cada rota alterada.

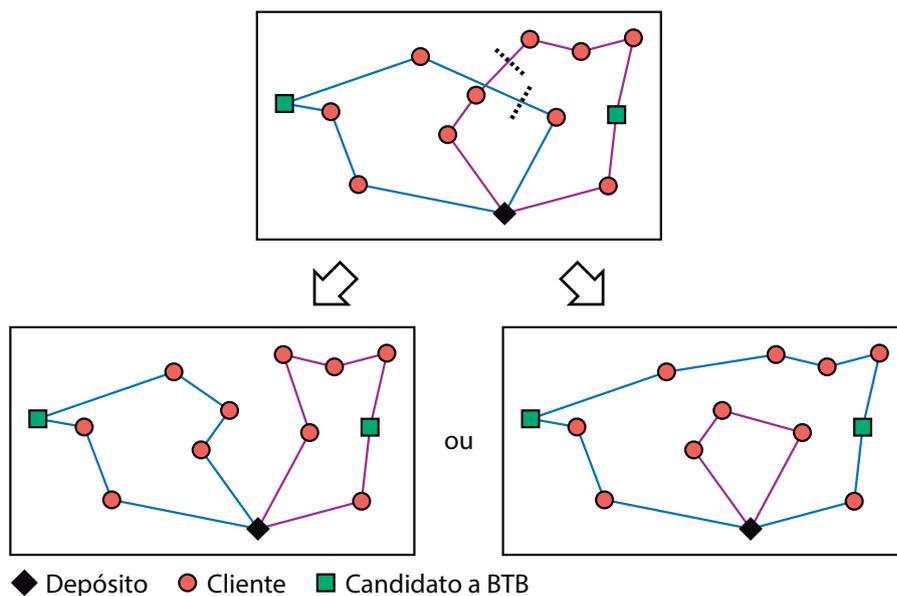


Figura 3.2: Demonstração da busca local *Cross* utilizada pelo RVND.

A busca *Shift*(1,0) (OSMAN, 1993) consiste em trocar um ponto de rota. Ela busca entre todos os pontos de todas as rotas aquele que, caso seja trocado para uma posição de outra rota, tenha o maior ganho para a solução.

A busca *Swap*(1,1) (OSMAN, 1993) troca a posição de dois pontos de rotas diferentes. Esta busca se resume em buscar a melhor combinação entre dois pontos de rotas diferentes que se trocados de posições alcance o maior ganho para a solução sem que ultrapasse a capacidade dos veículos das rotas alteradas.

Na busca *2-opt* (CROES, 1958) duas arestas não adjacentes são substituídas por duas novas arestas formando uma nova rota. Este algoritmo busca desconectar e reconectar duas arestas de uma mesma rota melhorando a solução final.

A busca *Reinsertion* (OR, 1976) reinsere um ponto em outro local da mesma rota. Ela busca o ponto em que movido para outra localização dentro da mesma rota tenha o maior ganho para a solução.

---

#### Algoritmo 2: RVND

---

**Require:** Solução parcial  $S$  para roteamento dos VEs;  
**Ensure:** Solução parcial  $S$  para roteamento dos VEs refinada;

- 1:  $melhora \leftarrow TRUE$ ;
- 2: **while**  $melhora$  **do**
- 3:    $i \leftarrow 0$ ;
- 4:    $buscas \leftarrow sorteioBL()$ ;
- 5:   **while**  $i < b$  **do**
- 6:      $S' \leftarrow rodaBL(S, i)$ ;
- 7:      $i \leftarrow i + 1$ ;
- 8:     **if**  $S'.custo < S.custo$  **then**
- 9:        $S \leftarrow S'$ ;
- 10:        $melhora \leftarrow TRUE$ ;
- 11:        $break$ ;
- 12:     **else**
- 13:        $melhora \leftarrow FALSE$ ;
- 14:     **end if**
- 15:   **end while**
- 16: **end while**
- 17: **return**  $S$ ;

---

A solução  $S$  é submetida a cada uma das buscas em uma ordem sorteada, como pode ser visto no Algoritmo 2. Caso haja alguma solução melhorada por qualquer das buscas, a iteração do RVND volta do início com a nova solução encontrada como solução de partida. A ordem em que as buscas são executadas é sorteada no início e a cada vez

que haja alguma melhora a solução.

Na Linha 4 do Algoritmo 2, *buscas* representa um vetor cujo tamanho se refere a quantidade de buscas utilizadas representada por *b*. Este vetor recebe a ordem na qual as buscas serão executadas recebida.

A função *rodaBL(S, i)*, na Linha 6, executa a *i*-ésima busca local do vetor que contém a ordem sorteada.

A iteração termina quando são executadas todas as buscas da lista na ordem sorteada sem que haja melhoria na solução.

### 3.1.3 Designação das estações de troca de bateria

Para se designar as ETBs que serão utilizadas pela solução, inicialmente, o Algoritmo 3 (IETB) recebe um valor inteiro *porcentagem* que representa o percentual da bateria que irá refletir em um raio de alcance que a ETB terá em relação aos clientes, representado no quadro 1 da Figura 3.2.

Na Linha 1 do Algoritmo 3 é inicializada a variável *S'* que irá receber a melhor solução encontrada. Das Linhas 2 a 9 é feita uma iteração com critério de parada sendo o valor de *porcentagem* atingir zero. Para isso, a cada iteração o valor de *porcentagem* é decrementado de 5 e para cada valor desse percentual da bateria, que gera um raio de alcance diferente, é criada uma solução que será comparada na Linha 5 com a anterior e mantida a melhor.

Para se obter as soluções a serem comparadas, primeiramente, na Linha 3 é utilizado um algoritmo baseado no conceito de Subconjunto Dominante Mínimo (SDM), descrito pelo Algoritmo 4, para que sejam pré selecionadas algumas das estações que possam melhor atender às rotas geradas anteriormente.

No Algoritmo 4, denominado SDM, é gerado um novo grafo que representa as ligações entre os pontos candidatos a ETBs e os clientes alcançados por eles baseados em uma porcentagem da bateria total dos veículos formando um raio em volta de cada candidato a estação, representado no quadro 1, 2 e 3 da Figura 3.2.

Com este novo grafo, encontra-se então um subconjunto dos pontos candidatos a ETBs em que tenha o menor somatório de custos de implantação e atendam todos os

**Algoritmo 3: IETB**


---

**Require:** Solução parcial  $S$  para roteamento dos VEs refinada e valor inteiro *porcentagem* inicial da bateria;  
**Ensure:** Solução  $S'$  para roteamento dos VEs e localização das ETBs;

- 1:  $S'.custo \leftarrow \infty$ ;
- 2: **while** *porcentagem* > 0 **do**
- 3:   *conjunto*  $\leftarrow$   $SDM(\textit{porcentagem})$ ;
- 4:    $S \leftarrow addETB(S, \textit{conjunto})$ ;
- 5:   **if**  $S.custo < S'.custo$  **then**
- 6:      $S' \leftarrow S$ ;
- 7:   **end if**
- 8:   *porcentagem*  $\leftarrow$  *porcentagem* - 5;
- 9: **end while**
- 10: **return**  $S'$ ;

---

clientes, vide quadros 4, 5 e 6 da Figura 3.2.

Na Linha 1 do Algoritmo 4, o conjunto  $CE$  recebe o conjunto  $J$  de todas as candidatas a estações de troca de bateria. A matriz de termos binários  $DM$ , na Linha 2, representa uma matriz de dominância  $ixj$ , onde suas linhas são representadas pelo conjunto de todos os clientes e as colunas pelas candidatas a ETBs. A célula  $DM[i][j]$  da matriz recebe 1 (um) quando, em relação ao raio obtido pela *porcentagem* utilizada, a candidata  $j$  consegue alcançar o cliente  $i$  e recebe 0 (zero) caso contrário. A matriz é preenchida pela função  $criaMatrizDominancia(\textit{porcentagem})$  que recebe o valor de *porcentagem* como parâmetro e, baseado na bateria do veículo, é calculado o raio de alcance.

Na Linha 3, o vetor  $VP$  representa o vetor prioridade criado pela função  $criaVetorPrioridade(DM)$ . Para cada candidato a estação  $j$ , o vetor recebe o somatório de todos os clientes que esta estação alcança (obtido na matriz  $DM$ ) dividido pelo quadrado do custo de instalação desta ETB, representado pela Equação 3.1.

$$VP[j] = \left( \sum_{i \in I} DM[i][j] \right) / (ETB[j].custo)^2, \quad \forall j \in J \quad (3.1)$$

Das Linhas 4 a 10 é feita uma iteração com critério de parada sendo o conjunto  $CE$  estar vazio. Na Linha 5 o *id\_maior* recebe o identificador da candidata a ETB com o maior valor no vetor  $VP$ .

A lista  $SDM\_ETB$  representa a lista das ETBs selecionadas pelo algoritmo  $SDM$

e na Linha 6 ela insere a candidata representada pelo  $id.maior$ . Em seguida, na Linha 7 o procedimento  $atualizaDominancias(DM, id.maior)$  atualiza a matriz DM colocando o valor 0 (zero) a todas as linhas representadas por todos os clientes atendidos pela ETB representada pelo  $id.maior$ . Desta forma, todos os clientes atendidos por esta estação selecionada não serão mais considerados pelas outras ETBs candidatas, vide quadro 4 da Figura 3.3.

Com a matriz  $DM$  atualizada, a função  $atualizaPrioridade(VP, DM)$  na Linha 8 atualiza também o vetor de prioridades  $VP$ , utilizando novamente a equação 3.1 para gerar os valores atualizados. Com esta atualização, alguns valores em  $VP$  poderão ser zerados, com isso, a função  $atualizaETB(CE, VP)$  na Linha 9 irá retirar do conjunto  $CE$  todas as candidatas a ETBs que tiverem o valor nulo no vetor  $VP$ .

Por fim, quando não houver mais nenhuma candidata no conjunto  $CE$ , o algoritmo termina e retorna o conjunto  $SDM\_ETB$ .

---

**Algoritmo 4: SDM**


---

**Require:** Valor *porcentagem* da bateria;  
**Ensure:** Subconjunto de candidatos a ETBs a ser utilizado pela solução;  
1:  $CE \leftarrow J$ ;  
2:  $DM \leftarrow criaMatrizDominancia(porcentagem)$ ;  
3:  $VP \leftarrow criaVetorPrioridade(DM)$ ;  
4: **while**  $|CE| > 0$  **do**  
5:    $id.maior \leftarrow maxPrioridade(VP)$ ;  
6:    $SDM\_ETB.insere(id.maior)$ ;  
7:    $atualizaDominancias(DM, id.maior)$ ;  
8:    $atualizaPrioridade(VP, DM)$ ;  
9:    $atualizaETB(CE, VP)$ ;  
10: **end while**  
11: **return**  $SDM\_ETB$ ;

---

Com o subconjunto de estações gerado pelo Algoritmo 4, volta-se à Linha 4 do Algoritmo 3 onde a função  $addETB(S, conjunto)$  irá utilizar deste subconjunto obtido para designar as ETBs necessárias. Para isso, é observado cada uma das rotas em busca de pontos de ruptura, pontos onde um veículo esgotaria sua bateria e não conseguiria alcançar o próximo ponto. Procura-se então alguma das estações candidatas existentes no subconjunto que atenda ao ponto mais próximo anteriormente àquela ruptura e esta estação é inserida na rota.

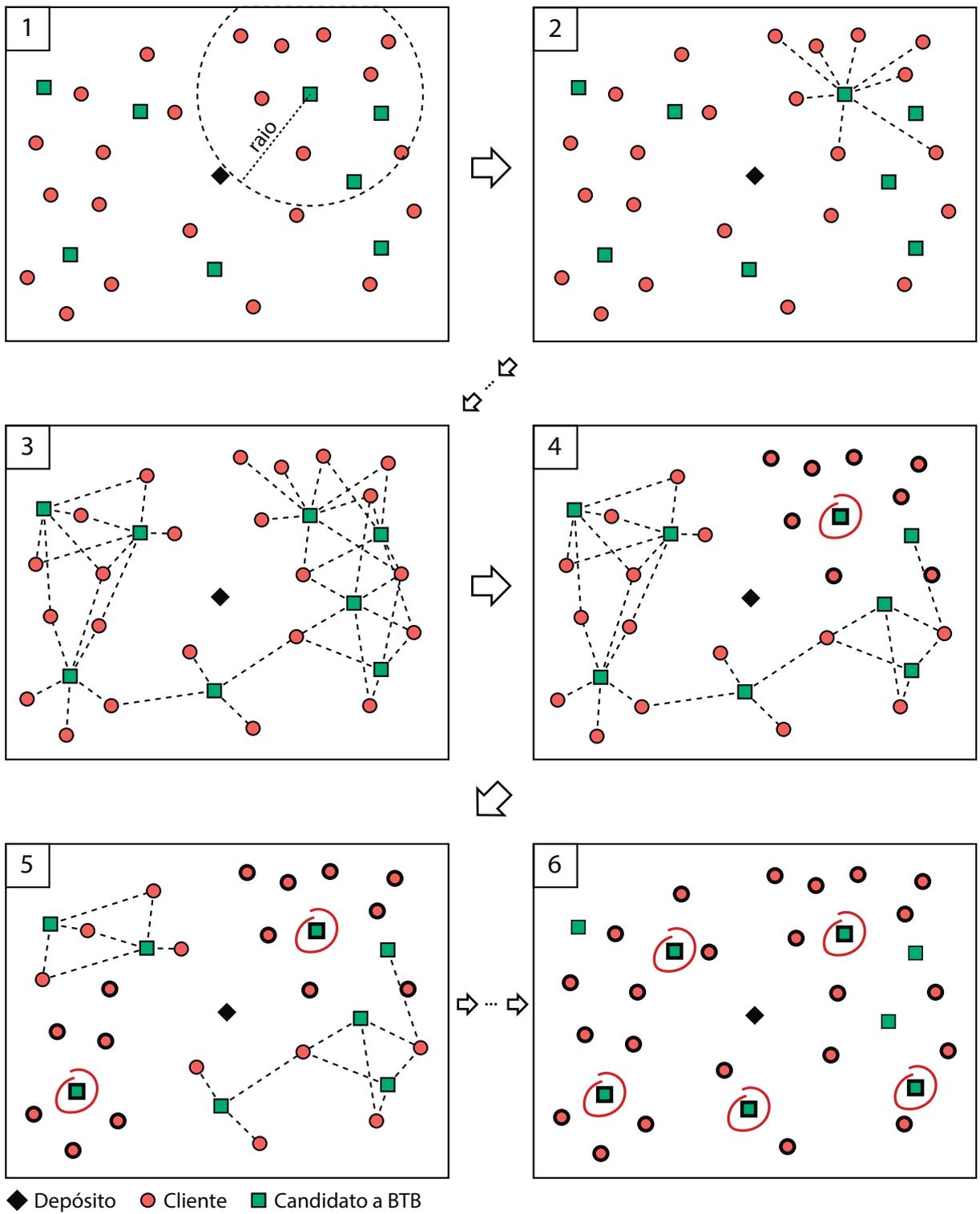


Figura 3.3: Demonstração do algoritmo SDM.

Caso não haja pontos no subconjunto que consiga atender a algum veículo em alguma ruptura, passa-se a adicionar a estação mais barata que atenda aquela rota mesmo que não pertença ao subconjunto. Esta ETB é adicionada ao subconjunto. A estação mais barata é aquela que possui o menor valor somando o custo de implantação daquela ETB com o valor da adição do caminho na rota.

No fim, quando não existe mais rupturas nas rotas, são excluídas as estações candidatas do subconjunto que não foram utilizados.

Para este experimento foi escolhido como valor de *porcentagem* da bateria para as iterações o valor inicial 40. No fim das iterações é retornada a solução com melhor resultado.

### 3.1.4 *Multi-start*

---

#### Algoritmo 5: Algoritmo IMB-SDM-RVND

---

**Require:** Quantidade de iterações sem melhora  $ISM$ , Valor *porcentagem* da bateria;  
**Ensure:** Solução para o problema de roteamento de VEs e de localização de ETBs;

- 1:  $i \leftarrow 0$ ;
- 2:  $S^*.custo \leftarrow \infty$ ;
- 3: **while**  $i < ISM$  **do**
- 4:    $S \leftarrow IMB()$ ;
- 5:    $S \leftarrow RVND(S)$ ;
- 6:    $S \leftarrow IETB(S, porcentagem)$ ;
- 7:    $i \leftarrow i + 1$ ;
- 8:   **if**  $S.custo < S^*.custo$  **then**
- 9:      $S^* \leftarrow S$ ;
- 10:     $i \leftarrow 0$ ;
- 11:   **end if**
- 12: **end while**
- 13: **return**  $S^*$ ;

---

Fazendo uso dos módulos descritos no algoritmo IMB-SDM-RVND, apresentado pelo Algoritmo 5, todos os processos são repetidos varias vezes e sempre armazenando a melhor solução  $S^*$ , semelhante à estrutura da meta-heurística GRASP (*Greedy Randomized Adaptive Search Procedure*)(FEO; RESENDE, 1995). Primeiramente é chamado o algoritmo  $IMB()$ , na Linha 4, que irá gerar as rotas iniciais para a solução, em seguida estas rotas serão refinadas pelo algoritmo  $RVND(S)$ , na Linha 5, e, por fim, o algoritmo

$IETB(S)$ , na Linha 6, se encarregará de designar as ETBs para a solução.

Para os testes, por haver aleatoriedades durante a heurística construtiva, foram utilizadas iterações consecutivas sem que haja melhora na solução, neste caso 50, representada pela variável  $ISM$  (iteraões sem melhora).

## 3.2 Abordagem Híbrida

A proposta desta abordagem é agregar técnicas de Programação Matemática à heurística anterior (Seção 3.1) fazendo refinamentos nas soluções obtidas, exercendo papel de busca local. Para isso, um segundo modelo foi usado e, com a utilização de um resolvedor, foi possível a agregação deste com a heurística.

### 3.2.1 Modelo por rotas

Neste modelo as variáveis de decisão estão relacionadas às possíveis rotas a serem utilizadas. O papel do modelo é selecionar, entre as rotas disponíveis, aquelas que melhor resolvem o problema.

A seguir o modelo é detalhado.

Definições:

- $R$  conjunto de rotas indexado por  $r$ ;
- $I$  conjunto dos clientes indexado por  $i$ .
- $J$  conjunto das ETBs candidatas indexado por  $j$ ;
- $B_r$  conjunto das ETBs utilizadas pela rota  $r$ .

Variáveis de decisão:

- $x_r$  variável de decisão binária para indicar se a rota  $r$  foi escolhida;
- $y_j$  variável de decisão binária para indicar se a ETB  $j$  foi selecionada.

Constantes:

- $c'_r$  custo da rota  $r$ ;
- $c''_j$  custo da ETB  $j$ ;
- $a_{ri}$  constante  $\{1, 0\}$  que indica se a rota  $r$  passa ou não pelo cliente  $i$ .

Modelo:

$$\text{Min } z = \sum_{r \in R} c'_r x_r + \sum_{j \in J} c''_j y_j \quad (3.2)$$

Sujeito a:

$$\sum_{r \in R} a_{ri} x_r \geq 1, \quad \forall i \in I \quad (3.3)$$

$$x_r \leq y_j, \quad \forall r \in R, \forall j \in B_r \quad (3.4)$$

$$x_r, y_j \in \{0, 1\}, \quad \forall i \in I, \forall j \in J \quad (3.5)$$

Neste modelo, a função objetivo (3.2) visa minimizar a soma dos custos das rotas e os custos das ETBs usadas. As restrições (3.3) garantem que todos os clientes são atendidos por pelo menos uma rota, enquanto as restrições (3.4) asseguram que se uma certa rota é utilizada, suas ETBs são designadas. As restrições (3.5) estabelecem a natureza binária das variáveis de decisão.

A maior dificuldade na utilização deste modelo é o elevado número de rotas possíveis. Por este motivo, neste trabalho será considerado apenas um subconjunto das rotas. Este subconjunto será composto por rotas construídas pelas heurísticas já apresentadas.

Comparando os resultados heurísticos como os resultados obtidos pela programação matemática, foi possível observar a dificuldade da heurística para identificar as melhores ETBs para um dado conjunto de rotas. Por conta disso, o objetivo do uso deste modelo é atuar exatamente neste ponto. As rotas obtidas pela heurística serão replicadas e diferentes combinações de ETBs serão consideradas.

### 3.2.2 *Geração de conjunto de rotas*

Considerando que cada solução é composta por um conjunto de rotas, para definir as rotas a serem incluídas no modelo, inicialmente esta solução é particionada, gerando um conjunto de rotas.

Além disso, para cada rota em que haja alguma ETB sendo utilizada, diferentes combinações de ETBs são consideradas como pode ser visto na Figura 3.4. Para isso é criada uma nova rota com BTB diferente e adicionada no ponto da rota que seja o mais

barato possível. Apenas as ETBs que o veículo consiga alcançar e que tornem a rota viável são consideradas. Caso haja mais de uma ETB na rota original, este número é considerado também para as variações.

Cada rota  $r$ , considerando a rota original e as rotas geradas, é relacionada a uma variável  $x_r$ . Além disso, para cada rota  $r$  é definido o conjunto  $B_r$  das ETBs utilizadas pela rota.

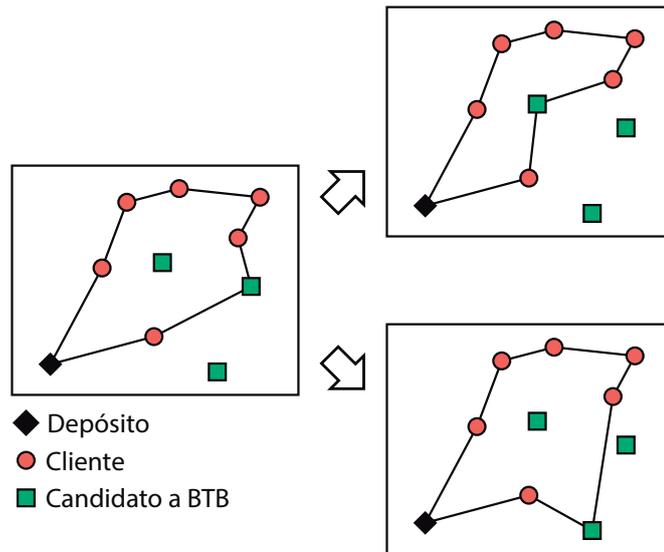


Figura 3.4: Exemplo de derivação de rotas

As novas rotas criadas com variações de ETBs são limitadas por apenas

### 3.2.3 Refinamento via Programação Matemática

Para viabilizar o uso do modelo por rotas para pós-refinamento, alguns ajustes foram feitos na heurística IMB\_SDM\_RVND, conforme apresentado no Algoritmo 6. A principal alteração é referente ao registro das  $n$  melhores soluções obtidas pela heurística. Cada uma destas soluções será refinada através do uso do modelo por rotas separadamente.

Além do número de iterações sem melhora (ISM), este algoritmo tem como parâmetro o número  $n$  de soluções (das melhores soluções) a serem armazenadas durante o processo de busca.

Como pode ser observada na primeira linha do Algoritmo 6, o primeiro passo é inicializar uma estrutura, denominada de Pool, capaz de armazenar as  $n$  melhores soluções. Os custos das soluções iniciais do Pool serão consideradas com valor muito grande para

que possam ser substituídas pelas soluções alcançadas pelo algoritmo.

Outra alteração foi na Linha 7 onde a qualidade da solução da iteração corrente é comparada com a qualidade da pior solução do Pool. Caso a solução seja melhor, a pior solução é removida do Pool (Linha 8) e a solução corrente é inserida na estrutura de forma a manter a estrutura ordenada (Linha 9);

Da Linha 14 a 23 são realizados os passos referentes ao refinamento via programação matemática. Na Linha 14 a melhor solução do Pool é atribuída a variável  $S^*$ , variável que armazena a melhor solução que será retornada pelo algoritmo. Em seguida, para cada solução  $i$  do Pool, na Linha 17, são geradas as rotas considerando as diferentes atribuições de ETBs. Na linha 18 é chamado o resolvedor que irá considerar o conjunto de rotas gerados a linha anterior além de utilizar como solução inicial, para o resolvedor, a própria solução  $i$  do Pool.

A solução obtida pelo modelo é comparada com a solução  $S^*$  na Linha 19. Caso a qualidade da solução encontrada pelo modelo seja melhor, a solução  $S^*$  é atualizada.

Na Linha 24 a solução  $S^*$  é verificada para se concluir que nenhum cliente possa ter sido atendido mais de uma vez por rotas distintas. Caso isso ocorra, este cliente é retirado da rota em que o impacto ao resultado seja o melhor possível, caso contrário, a solução é mantida. Por fim, na Linha 25, a melhor solução é retornada.

Neste trabalho, após testes empíricos, foi considerado  $n = 4$ .

---

**Algoritmo 6:** Algoritmo IMB-SDM-RVND-PL
 

---

**Require:** Quantidade de iterações sem melhora  $ISM$ , número  $n$  de soluções do Pool;

**Ensure:** Solução para o problema de roteamento de VEs e de localização de ETBs melhorada;

- 1: *inicializa*(*Pool*,  $n$ );
- 2:  $i \leftarrow 0$ ;
- 3: **while**  $i < ISM$  **do**
- 4:    $S \leftarrow IMB()$ ;
- 5:    $S \leftarrow RVND(S)$ ;
- 6:    $S \leftarrow IETB(S)$ ;
- 7:   **if**  $S.custo < Pool.pior.custo$  **then**
- 8:      $Pool.removePior()$ ;
- 9:      $Pool.insereOrdenado(S)$ ;
- 10:     $i \leftarrow 0$ ;
- 11:   **end if**
- 12:    $i \leftarrow i + 1$ ;
- 13: **end while**
- 14:  $S^* \leftarrow Pool.melhorSolucao()$ ;
- 15:  $i \leftarrow 0$ ;
- 16: **while**  $i < n$  **do**
- 17:    $rotas \leftarrow geraRotas(Pool.solucao(i))$ ;
- 18:    $S \leftarrow resovedor(Pool.solucao(i), rotas)$ ;
- 19:   **if**  $S.custo < S^*.custo$  **then**
- 20:      $S^* \leftarrow S$ ;
- 21:   **end if**
- 22:    $i \leftarrow i + 1$ ;
- 23: **end while**
- 24:  $S^* \leftarrow verificaSolucao(S^*)$ ;
- 25: **return**  $S^*$ ;

---

## 4 Resultados computacionais

Os experimentos computacionais foram executados com o algoritmo proposto implementado em C++, usando também o resolvidor CPLEX e executados em uma máquina virtual utilizando o VMWare com 8 vCPU (Intel® Xeon(R) CPU E7530 @ 1.87GHz), 16GB de RAM e com sistema operacional Linux Ubuntu 16.04.1 LTS.

### 4.1 Instâncias e parâmetros

Para este estudo, foram utilizadas algumas das instâncias de CVRP (problema de roteamento de veículos capacitados) modificadas por Yang e Sun (2015) para se adaptarem ao problema proposto. Todas as instâncias utilizadas podem ser obtidas em <http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/>.

As instâncias selecionadas são do conjunto de pequeno porte (AUGERAT et al., 1998) que apresentam de 16 a 70 clientes, de médio porte (ROCHAT; TAILLARD, 1995) com 75 a 150 clientes e duas instâncias de grande porte (GOLDEN et al., 1998) com 255 e 480 clientes.

O quarto conjunto de instâncias, de tamanhos muito pequenos, são utilizadas para comparação com o resolvidor CPLEX, por este ser incapaz de resolver instâncias maiores em tempo viável. Estas são derivados da instância P-n16-k8 de Augerat et al. (1998).

As modificações foram feitas pelos seguintes parâmetros, descritos em Hof, Schneider e Goeke (2017):

- Para o conjunto de Augerat et al. (1998) o depósito está localizado em (1, -1). Para as outras instâncias o depósito é o primeiro vértice da lista;
- Em todos os conjuntos de instâncias, cada localização de clientes representa também um ponto candidato para uma ETB;
- A capacidade da bateria  $B$  é  $1,2d_{max}$  e arredondada para o valor inteiro mais

próximo, onde  $d_{max}$  corresponde à distância máxima entre quaisquer dois vértices da instância;

- O custo de construção de todas as ETBs é definido como  $0,5B$  e arredondado para o valor inteiro mais próximo;
- As instâncias muito pequenas para solução com o modelo matemático são derivadas da instância P-n16-k8 de Augerat et al. (1998) mantendo os últimos  $n = 6, 7$  e  $8$  clientes da instância e removendo os restantes. Além disso, a capacidade de carga foi modificada para 40.

Além dos conjuntos de instâncias apresentadas utilizadas também pela literatura, para este estudo foram criadas variações destes arquivos, com as marcas "D" e "R", onde os valores de instalação das ETB fossem diferentes para cada ponto candidato, diferentes das instâncias anteriormente propostas onde os custos das estações são os mesmos para qualquer dos pontos candidatos. Os arquivos "D" são instâncias onde o custo de cada ETB é baseado na densidade de clientes ao redor do candidato em questão, sendo mais caro implantar uma ETB em um candidato que possua mais clientes em sua vizinhança próxima. Já as instâncias marcadas com "R" são aquelas onde os valores de instalação de cada ETB é gerado aleatoriamente. Estas novas instâncias foram descritas em Portela, Gonçalves e Soares (2017).

## 4.2 Resultados

Neste estudo, além de uma comparação com a literatura, foi apresentado um comparativo entre a abordagem puramente heurística IMB-SDM-RVND com a abordagem híbrida IMB-SDM-RVND-PL onde é feito um refinamento usando o modelo auxiliar.

Primeiramente, na Tabela 4.1, foi feita uma comparação entre os resultados obtidos pelo CPLEX com tempo limite de 108.000 segundos (30 horas) com os resultados dos algoritmos IMB-SDM-RVND e IMB-SDM-RVND-PL propostos neste trabalho, utilizando instâncias da literatura e modificações das mesmas, como descrito em Portela, Gonçalves e Soares (2017). Para esta tabela é representada para cada instância, além do

nome,  $|I|$  que representa o número de clientes a serem atendidos,  $|K|$  que representa o número de veículos utilizados e  $|J|$  que representa a quantidade de ETBs utilizadas pela solução. Para o CPLEX são representados o custo da solução ótima  $f$  e o tempo  $t$  de execução em segundos. Para os demais algoritmos são representados o valor do melhor resultado obtido entre as dez execuções, o tempo médio  $t^m$  das execuções em segundos e, por fim, a diferença percentual  $\Delta(\%)$  que indica quão distante estão os resultados das heurísticas em relação à solução ótima obtida via CPLEX. Os resultados destacados em negrito representam os melhores resultados obtidos.

INPUTS				CPLEX		IMB-SDM-RVND			IMB-SDM-RVND-PL		
Instance	$ I $	$ K $	$ J $	$f$	$t(s)$	$f$	$t^m(s)$	$\Delta(\%)$	$f$	$t^m(s)$	$\Delta(\%)$
P-n6-k2	6	2	2	<b>426,86</b>	10,51	<b>426,86</b>	0,04	0,00	<b>426,86</b>	1,75	0,00
P-n7-k3	7	2	2	<b>428,60</b>	489,69	<b>428,60</b>	0,05	0,00	<b>428,60</b>	2,22	0,00
P-n8-k3	8	3	2	<b>597,16</b>	1853,04	<b>597,16</b>	0,07	0,00	<b>597,16</b>	1,82	0,00
P-n6-k2-D	6	2	2	<b>412,20</b>	279,74	<b>412,20</b>	0,05	0,00	<b>412,20</b>	2,01	0,00
P-n7-k3-D	7	2	2	<b>413,94</b>	2606,41	<b>413,94</b>	0,05	0,00	<b>413,94</b>	1,89	0,00
P-n8-k3-D	8	3	2	<b>585,93</b>	104805,02	595,16	0,08	1,57	<b>585,93</b>	1,73	0,00
P-n6-k2-R	6	2	1	<b>425,50</b>	368,19	430,20	0,04	1,11	<b>425,50</b>	2,02	0,00
P-n7-k3-R	7	2	2	<b>411,94</b>	2263,16	<b>411,94</b>	0,05	0,00	<b>411,94</b>	1,60	0,00
P-n8-k3-R	8	3	2	<b>548,16</b>	52250,34	<b>548,16</b>	0,08	0,00	<b>548,16</b>	2,01	0,00
Avg.				472,25	18325,12	473,80	0,06	0,30	472,25	1,89	0,00

Tabela 4.1: Comparativo dos resultados entre CPLEX, IMB-SDM-RVND e IMB-SDM-RVND-PL para as instâncias de tamanho muito pequenas.

Na Tabela 4.1 é possível observar que o algoritmo IMB-SDM-RVND consegue em quase todas as instâncias testadas alcançar o resultado ótimo obtido pelo CPLEX com tempo médio significativamente baixos, 0,06s. Em compensação, o algoritmo IMB-SDM-RVND-PL consegue obter o resultado ótimo para todas as instâncias testadas, vide os valores  $\Delta(\%)$ , mesmo que com o tempo médio um pouco maior, 1,89s, mas ainda bem baixo se comparado ao tempo gasto pelo resolvedor, sendo observado na média final como sendo cerca de 0,01% do tempo do CPLEX.

A Tabela 4.2 mostra comparações para algumas das instâncias de pequeno e médio porte, com custo de implantação de ETBs fixos e variados, entre as duas versões do algoritmo apresentados neste trabalho, IMB-SDM-RVND e IMB-SDM-RVND-PL. Para

INPUTS		IMB-SDM-RVND				IMB-SDM-RVND-PL				
Instance	$ I $	$ K $	$ J $	$f$	$t^m(s)$	$ K $	$ J $	$f$	$t^m(s)$	$\Delta(\%)$
P-n16-k8	16	8	2	1341,58	0,29	8	1	<b>1301,71</b>	0,78	-2,97
P-n19-k2	19	2	1	<b>467,03</b>	0,33	2	1	<b>467,03</b>	1,53	0,00
P-n21-k2	21	2	1	475,27	0,33	2	1	<b>474,88</b>	2,40	-0,08
P-n23-k8	23	8	3	1437,68	0,59	8	1	<b>1350,30</b>	2,31	-6,08
P-n45-k5	45	5	3	958,61	3,19	5	2	<b>913,75</b>	8,17	-4,68
P-n70-k10	70	10	5	1857,61	13,49	10	2	<b>1690,80</b>	28,25	-8,98
tai75b	75	10	1	<b>1480,53</b>	15,75	10	1	<b>1480,53</b>	15,99	0,00
tai100d	100	12	0	<b>1684,99</b>	35,54	12	0	<b>1684,99</b>	35,80	0,00
tai150a	150	15	1	3365,43	117,23	15	1	<b>3344,88</b>	117,86	-0,61
Avg.				1452,08	20,75			1412,10	23,68	-2,60
P-n16-k8-D	16	8	3	1370,72	0,30	8	1	<b>1305,09</b>	1,49	-4,79
P-n19-k2-D	19	2	1	<b>495,03</b>	0,35	2	1	<b>495,03</b>	2,10	0,00
P-n21-k2-D	21	2	1	<b>491,84</b>	0,37	2	1	<b>491,84</b>	2,33	0,00
P-n23-k8-D	23	8	2	1467,82	0,60	8	1	<b>1351,47</b>	1,05	-7,93
P-n45-k5-D	45	5	3	1001,30	3,57	5	2	<b>962,49</b>	6,69	-3,88
P-n70-k10-D	70	10	7	1749,94	11,63	10	3	<b>1619,56</b>	42,87	-7,45
tai75b-D	75	10	1	<b>1400,75</b>	15,21	10	1	<b>1400,75</b>	15,47	0,00
tai100d-D	100	12	1	<b>1677,28</b>	37,93	12	1	<b>1677,28</b>	38,69	0,00
tai150a-D	150	15	1	<b>3273,45</b>	106,78	15	1	<b>3273,45</b>	108,02	0,00
Avg.				1436,46	19,64			1397,44	24,30	-2,67
P-n16-k8-R	16	8	3	1280,29	0,38	8	2	<b>1258,79</b>	2,60	-1,68
P-n19-k2-R	19	2	2	447,85	0,26	2	1	<b>446,32</b>	2,20	-0,34
P-n21-k2-R	21	2	2	<b>458,48</b>	0,40	2	2	<b>458,48</b>	3,15	0,00
P-n23-k8-R	23	8	2	1441,82	0,67	8	2	<b>1336,71</b>	2,02	-7,29
P-n45-k5-R	45	5	3	965,85	3,74	5	2	<b>900,99</b>	6,49	-6,72
P-n70-k10-R	70	11	4	1999,22	9,37	10	2	<b>1685,65</b>	16,46	-15,68
tai75b-R	75	10	1	1480,18	17,84	10	1	<b>1476,10</b>	18,08	-0,28
tai100d-R	100	12	1	<b>1660,75</b>	31,95	12	1	<b>1660,75</b>	32,72	0,00
tai150a-R	150	15	1	3267,33	92,59	15	1	<b>3259,18</b>	93,56	-0,25
Avg.				1444,64	17,47			1387,00	19,70	-3,58

Tabela 4.2: Comparativo dos resultados entre IMB-SDM-RVND e IMB-SDM-RVND-PL para as instâncias de pequeno e médio porte.

esta tabela, assim como para a anterior,  $|I|$ ,  $|K|$  e  $|J|$  representam, respectivamente, o número de clientes da instância, o número de veículos e a quantidade de ETBs utilizados pela solução. São representados o valor do melhor resultado da função  $f$  obtido entre dez execuções, o tempo médio  $t^m$  das execuções em segundos e, por fim, a diferença percentual  $\Delta(\%)$  entre as soluções comparadas. Os resultados destacados em negrito representam os melhores resultados obtidos.

Na Tabela 4.2 é nítida a contribuição do modelo que compõe o método híbrido, representado pelo algoritmo IMB-SDM-RVND-PL, mostrando um ganho nos valores de quase todas as soluções testadas. Como o modelo trata-se basicamente de um refinamento nas distribuições de ETBs, pode-se observar que o ganho se dá em grande parte pela diminuição (expressiva em algumas instâncias) da quantidade de estações ( $|J|$ ) utilizadas pelas soluções. Em contrapartida, o aumento do tempo médio não representa um impacto significativo. Com isso, a abordagem híbrida mostra ser de grande ajuda à abordagem utilizada.

Ainda na Tabela 4.2 é notado que para instâncias que utilizam de 1 ou nenhum ETB, o tempo acrescentado pela utilização do modelo é muito pequena já que este não receberia uma solução com muitas chances de melhoria em relação as disposições das ETBs e com isso o modelo não teria muito trabalho, mantendo até o mesmo resultado do alcançado apenas pela heurística pura.

Na Tabela 4.3, é apresentado um comparativo entre a abordagem híbrida proposta neste trabalho com a abordagem AVNS proposta em Hof, Schneider e Goeke (2017). Novamente, para esta tabela, assim como para as anteriores,  $|I|$ ,  $|K|$  e  $|J|$  representam, respectivamente, o número de clientes da instância, o número de veículos e a quantidade de ETBs utilizados pela solução. São representados o valor do melhor resultado da função  $f$  obtido entre dez execuções, o tempo médio  $t^m$  das execuções em segundos e, por fim, a diferença percentual  $\Delta(\%)$  representa uma comparação entre as soluções do algoritmo IMB-SDM-RVND-PL, deste trabalho, com as da abordagem AVNS da literatura. Os resultados destacados em negrito representam os melhores resultados na comparação.

Por fim, a Tabela 4.4 mostra o comparativo entre a abordagem híbrida proposta com a abordagem SIGALNS proposta em Yang e Sun (2015). Para esta tabela, além

INPUTS		AVNS				IMB-SDM-RVND-PL				
Instance	$ I $	$ K $	$ J $	$f$	$t^m(s)$	$ K $	$ J $	$f$	$t^m(s)$	$\Delta(\%)$
P-n16-k8	16	8	1	<b>1282,38</b>	1,66	8	1	1301,71	0,78	1,51
P-n19-k2	19	2	1	468,08	1,98	2	1	<b>467,03</b>	1,53	-0,22
P-n21-k2	21	2	1	<b>472,74</b>	1,93	2	1	474,88	2,40	0,45
P-n23-k8	23	8	1	<b>1347,04</b>	4,15	8	1	1350,30	2,31	0,24
P-n40-k5	40	5	1	<b>857,70</b>	8,42	5	1	877,56	5,03	2,32
P-n45-k5	45	5	1	<b>872,23</b>	12,12	5	2	913,75	8,17	4,76
P-n50-k7	50	7	2	<b>1130,44</b>	15,18	7	2	1156,60	14,40	2,31
P-n55-k8	55	7	2	<b>1170,04</b>	16,25	7	2	1212,37	26,56	3,62
P-n60-k10	60	10	2	<b>1556,38</b>	33,57	10	2	1598,99	20,41	2,74
P-n70-k10	70	10	2	<b>1632,87</b>	21,30	10	2	1690,80	28,25	3,55
tai75a	75	10	0	<b>1664,08</b>	8,84	11	0	1713,21	15,79	2,95
tai75b	75	10	1	<b>1471,57</b>	12,1	10	1	1480,53	15,99	0,61
tai75c	75	9	0	<b>1381,20</b>	18,11	9	0	1421,02	16,31	2,88
tai75d	75	9	0	<b>1405,45</b>	5,44	10	0	1410,58	12,69	0,37
tai100a	100	12	0	<b>2179,28</b>	19,79	12	1	2233,71	31,30	2,50
tai100b	100	11	0	<b>1948,73</b>	10,46	11	0	1954,73	37,35	0,31
tai100c	100	11	1	1598,72	27,76	11	1	<b>1598,33</b>	42,59	-0,02
tai100d	100	11	0	<b>1609,06</b>	20,80	12	0	1684,99	35,80	4,72
tai150a	150	15	0	<b>3194,41</b>	58,89	15	1	3344,88	117,86	4,71
tai150b	150	14	0	<b>2815,80</b>	45,23	14	0	2820,15	146,26	0,15
tai150c	150	15	0	<b>2403,41</b>	43,44	15	0	2445,58	121,69	1,75
tai150d	150	14	0	<b>2744,54</b>	84,91	15	1	2852,90	127,89	3,95
GWKC09	255	14	3	<b>666,05</b>	1017,08	14	3	698,21	3197,08	4,83
GWKC16	480	37	0	<b>1693,34</b>	262,64	37	1	1792,85	3199,52	5,88
Avg.				1565,23	73,00			1603,99	301,16	2,37

Tabela 4.3: Comparativo dos resultados entre a abordagem híbrida IMB-SDM-RVND-PL e a abordagem AVNS.

das informações exibidas na Tabela 4.3, é apresentada a média dos resultados  $f^m$  obtidos no experimento feito 10 vezes além do melhor resultado  $f$ . Esta média também pode ser comparada ao da literatura. Os resultados destacados em negrito representam os melhores resultados na comparação.

Pode-se observar que os resultados obtidos pela abordagem proposta neste trabalho se encontram bem próximos dos resultados da literatura, publicados em Hof, Schneider e Goeke (2017), chegando a conseguir melhorias em duas das instâncias utilizadas e com um tempo não muito maior. Caso sejam comparados com os obtidos pelo algoritmo SIGALNS, na publicação onde o problema estudado foi apresentado (YANG; SUN, 2015), pode-se observar ganhos bastante expressivos em quase todas as instâncias e com tempos bem melhores.

INPUTS	SIGALNS				IMB-SDM-RVND + PL				$\Delta(\%)$			
	$ I $	$ K $	$ J $	$f$	$f^m$	$\#^m(s)$	$ K $	$ J $		$f$	$f^m$	$\#^m(s)$
P-n16-k8	16	8	1	<b>1281,95</b>	1299,47	2,58	8	1	1301,71	1301,71	0,78	1,54
P-n19-k2	19	2	1	471,39	483,95	2,78	2	1	<b>467,03</b>	469,66	1,53	-0,92
P-n21-k2	21	2	1	478,64	478,95	3,13	2	1	<b>474,88</b>	475,63	2,40	-0,79
P-n23-k8	23	9	1	1360,51	1436,20	3,14	8	1	<b>1350,30</b>	1399,86	2,31	-0,75
P-n40-k5	40	5	1	893,23	908,05	6,18	5	1	<b>877,56</b>	897,26	5,03	-1,75
P-n45-k5	45	5	2	939,63	950,39	7,69	5	2	<b>913,75</b>	924,64	8,17	-2,75
P-n50-k7	50	7	2	1196,48	1221,07	8,52	7	2	<b>1156,60</b>	1173,85	14,40	-3,33
P-n55-k8	55	7	2	1247,10	1268,30	20,13	7	2	<b>1212,37</b>	1226,46	26,56	-2,78
P-n60-k10	60	10	3	1684,24	1687,65	24,50	10	2	<b>1598,99</b>	1614,92	20,41	-5,06
P-n70-k10	70	10	3	1738,98	1764,56	35,93	10	2	<b>1690,80</b>	1713,68	28,25	-2,77
tai75a	75	10	4	1924,32	1967,90	53,69	11	0	<b>1713,21</b>	1782,45	15,79	-10,97
tai75b	75	10	3	1607,22	1658,93	76,73	10	1	<b>1480,53</b>	1488,31	15,99	-7,88
tai75c	75	9	4	1602,15	1654,52	75,11	9	0	<b>1421,02</b>	1482,71	16,31	-11,31
tai75d	75	10	3	1643,63	1647,98	51,06	10	0	<b>1410,58</b>	1468,32	12,69	-14,18
tai100a	100	12	4	2467,90	2509,95	118,80	12	1	<b>2233,71</b>	2258,73	31,30	-9,49
tai100b	100	13	5	2393,34	2431,90	134,42	11	0	<b>1954,73</b>	1964,71	37,35	-18,33
tai100c	100	11	4	1783,45	1834,85	123,29	11	1	<b>1598,33</b>	1616,86	42,59	-10,38
tai100d	100	12	4	1926,96	1986,75	188,95	12	0	<b>1684,99</b>	1742,27	35,80	-12,56
tai150a	150	15	4	3620,34	3722,45	329,67	15	1	<b>3344,88</b>	3399,74	117,86	-7,61
tai150b	150	14	7	3354,00	3375,64	367,24	14	0	<b>2820,15</b>	2875,02	146,26	-15,92
tai150c	150	15	5	2879,32	2958,83	337,44	15	0	<b>2445,58</b>	2464,85	121,69	-15,06
tai150d	150	15	5	3121,36	3175,92	492,06	15	1	<b>2852,90</b>	2878,94	127,89	-8,60
GWKC09	255	14	3	790,99	814,64	1798,23	14	3	<b>698,21</b>	708,29	3197,08	-11,73
GWKC16	480	38	12	2359,08	2397,72	10695,97	37	1	<b>1792,85</b>	1809,43	3199,52	-24,00
Avg.	101,42	10,67	3,50	1781,93	1818,19	623,22	10,83	1,00	1603,99	1630,76	301,16	-8,22

Tabela 4.4: Comparativo dos resultados entre a abordagem híbrida IMB-SDM-RVND-PL e a abordagem SIGNALNS.

## 5 Conclusões e trabalhos futuros

Este trabalho teve como objetivo estudar e propor novas abordagens para buscar boas soluções para o Problema de Roteamento de Veículos e Elétricos com Localização de Estações de Troca de Baterias. Este problema foi proposto com objetivo de viabilizar a utilização de veículos elétricos na área de logística, surgindo como uma das linhas de pesquisas de Logística Verde com o intuito de, além de reduzir gastos, como na maioria dos problemas clássicos, veio como uma alternativa para redução de impactos ambientais causados por automóveis movidos a combustão.

Neste trabalho foi apresentada uma estratégia, primeiro puramente heurística e posteriormente uma híbrida, em que se foi utilizado de programação matemática através de um modelo para integrar à abordagem heurística. Foi utilizada uma inspiração combinada dos Algoritmos de Inserção Mais Barata para montagem das rotas, do conceito de Subconjunto Dominante Mínimo para auxiliar na designação das estações, do método RVND para refinamento das rotas e Programação Matemática para refinamento das estações designadas formando uma abordagem híbrida para o problema.

Os experimentos foram realizados a partir de dados gerados pela literatura e também por modificações propostas neste trabalho.

Pelos resultados obtidos é notória a contribuição que o refinamento feito pelo método híbrido traz aos resultados sem um impacto tão grande ao tempo de execução, como visto nas tabelas de resultados. Pode-se observar o ganho expressivo em algumas instâncias, tanto nos valores finais, quanto na quantidade de ETBs utilizadas, principalmente para as instâncias modificadas, com valores de implantação de ETBs diferentes entre si, mostrada na Tabela 4.2.

A abordagem apresentada mostra-se competitiva em relação as apresentadas na literatura, chegando a ser melhor na maioria dos casos para a apresentada em Yang e Sun (2015) e também sendo melhor em duas das instâncias apresentadas em Hof, Schneider e Goeke (2017).

Para trabalhos futuros há o objetivo de buscar o aperfeiçoamento da abordagem

---

apresentada para resolver este problema com cada vez mais eficiência, independente dos custos de implementação de ETBs iguais ou diferentes e tentando se aproximar mais ainda dos resultados da literatura. Para isso é planejado a utilização de um algoritmo iterativo, alternando refinamento via RVND para rotas e Modelo para inserção de ETBs. Além disso, serão feitas execuções com as novas instâncias propostas em Hof, Schneider e Goeke (2017) em que as localizações de ETBs candidatas se diferem das localizações de clientes, diferentemente das instâncias usadas até então.

## Bibliografia

- AUGERAT, P. et al. *Computational results with a branch-and-cut code for the capacitated vehicle routing problem*. 1998.
- BARAN, R.; LEGEY, L. F. L. Veículos elétricos: história e perspectivas no brasil. *BNDES Setorial*, n. 33, mar. 2011, p. 207-224, Banco Nacional de Desenvolvimento Econômico e Social, 2011.
- BERMAN, O.; LARSON, R. C.; FOUSKA, N. Optimal location of discretionary service facilities. *Transportation Science*, INFORMS, v. 26, n. 3, p. 201–211, 1992.
- CASTRO, B. H. R. d.; FERREIRA, T. T. Veículos elétricos: aspectos básicos, perspectivas e oportunidades. *BNDES Setorial*, n. 32, set. 2010, p. 267-310, Banco Nacional de Desenvolvimento Econômico e Social, 2010.
- CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, Inform, v. 12, n. 4, p. 568–581, 1964.
- CROES, G. A. A method for solving traveling-salesman problems. *Operations research*, INFORMS, v. 6, n. 6, p. 791–812, 1958.
- DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. *Management science*, Inform, v. 6, n. 1, p. 80–91, 1959.
- FEO, T. A.; RESENDE, M. G. Greedy randomized adaptive search procedures. *Journal of global optimization*, Springer, v. 6, n. 2, p. 109–133, 1995.
- GOLDEN, B. L. et al. The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results. In: *Fleet management and logistics*. [S.l.]: Springer, 1998. p. 33–56.
- HOCHBAUM, D. S. (Ed.). *Approximation Algorithms for NP-hard Problems*. Boston, MA, USA: PWS Publishing Co., 1997. ISBN 0-534-94968-1.
- HODGSON, M. J. A flow-capturing location-allocation model. *Geographical Analysis*, Wiley Online Library, v. 22, n. 3, p. 270–279, 1990.
- HOF, J.; SCHNEIDER, M.; GOEKE, D. Solving the battery swap station location-routing problem with capacitated electric vehicles using an avns algorithm for vehicle-routing problems with intermediate stops. *Transportation Research Part B: Methodological*, Elsevier, v. 97, p. 102–112, 2017.
- KUBY, M.; LIM, S. Location of alternative-fuel stations using the flow-refueling location model and dispersion of candidate sites on arcs. *Networks and Spatial Economics*, Springer, v. 7, n. 2, p. 129–152, 2007.
- LAPORTE, G.; NOBERT, Y. An exact algorithm for minimizing routing and operating costs in depot location. *European Journal of Operational Research*, Elsevier, v. 6, n. 2, p. 224–226, 1981.

- LAPORTE, G.; NOBERT, Y.; ARPIN, D. An exact algorithm for solving a capacitated location-routing problem. *Annals of Operations Research*, Springer, v. 6, n. 9, p. 291–310, 1986.
- LAPORTE, G.; NOBERT, Y.; TAILLEFER, S. Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation science*, INFORMS, v. 22, n. 3, p. 161–172, 1988.
- LIN, S.-W. et al. Applying hybrid meta-heuristics for capacitated vehicle routing problem. *Expert Systems with Applications*, Elsevier, v. 36, n. 2, p. 1505–1512, 2009.
- LIPOWSKI, A.; LIPOWSKA, D. Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications*, Elsevier, v. 391, n. 6, p. 2193–2196, 2012.
- LORENA, L. A. N. Análise de redes. *Análise Espacial de Dados Geográficos*, INPE São José dos Campos, 2001.
- LYSGAARD, J.; LETCHFORD, A. N.; EGGLESE, R. W. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, Springer, v. 100, n. 2, p. 423–445, 2004.
- MIN, H.; JAYARAMAN, V.; SRIVASTAVA, R. Combined location-routing problems: A synthesis and future research directions. *European Journal of Operational Research*, Elsevier, v. 108, n. 1, p. 1–15, 1998.
- MIRCHANDANI, P.; ADLER, J.; MADSEN, O. B. New logistical issues in using electric vehicle fleets with battery exchange infrastructure. *Procedia-Social and Behavioral Sciences*, Elsevier, v. 108, p. 3–14, 2014.
- NAGY, G.; SALHI, S. Location-routing: Issues, models and methods. *European Journal of Operational Research*, Elsevier, v. 177, n. 2, p. 649–672, 2007.
- OR, I. Traveling salesman problem-type combinatorial optimization problems and their relation to the logistics of regional blood banking. *Northwestern University, Evanston, IL*, 1976.
- OSMAN, I. H. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of operations research*, Springer, v. 41, n. 4, p. 421–451, 1993.
- PANG, C. et al. Dominating sets in directed graphs. *Information Sciences*, Elsevier, v. 180, n. 19, p. 3647–3652, 2010.
- PORTELA, B. M. O.; GONçALVES, L. B.; SOARES, S. S. R. F. Abordagem heurística para o problema de roteamento de veículos elétricos e de localização de estações de troca de baterias com custos variados. *CBIC, Rio de Janeiro*, 2017.
- ROCHAT, Y.; TAILLARD, É. D. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of heuristics*, Springer, v. 1, n. 1, p. 147–167, 1995.
- ROSENKRANTZ, D. J.; STEARNS, R. E.; LEWIS, P. M. Approximate algorithms for the traveling salesperson problem. In: *15th Annual Symposium on Switching and Automata Theory (swat 1974)*. [S.l.: s.n.], 1974. p. 33–42. ISSN 0272-4847.

- SBIHI, A.; EGGLESE, R. W. Combinatorial optimization and green logistics. *4OR*, Springer, v. 5, n. 2, p. 99–116, 2007.
- SBIHI, A.; EGGLESE, R. W. Combinatorial optimization and green logistics. *Annals of Operations Research*, Springer, v. 175, n. 1, p. 159–175, 2010.
- SUBRAMANIAN, A. Heuristic, exact and hybrid approaches for vehicle routing problems. *Universidade Federal Fluminense. Tese de doutorado. Niterói*, p. 13, 2012.
- TAILLARD, É. et al. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science*, INFORMS, v. 31, n. 2, p. 170–186, 1997.
- TOTH, P.; VIGO, D. The vehicle routing problem, Philadelphia: society for industrial and applied mathematics. 2002.
- TOTH, P.; VIGO, D. *Vehicle routing: problems, methods, and applications*. [S.l.]: Siam, 2014. v. 18.
- WU, T.-H.; LOW, C.; BAI, J.-W. Heuristic solutions to multi-depot location-routing problems. *Computers & Operations Research*, Elsevier, v. 29, n. 10, p. 1393–1415, 2002.
- YANG, J.; SUN, H. Battery swap station location-routing problem with capacitated electric vehicles. *Computers & Operations Research*, Elsevier, v. 55, p. 217–232, 2015.
- ZARANDI, M. H. F.; HEMMATI, A.; DAVARI, S. The multi-depot capacitated location-routing problem with fuzzy travel times. *Expert Systems with Applications*, Elsevier, v. 38, n. 8, p. 10075–10084, 2011.
- ZARANDI, M. H. F. et al. Capacitated location-routing problem with time windows under uncertainty. *Knowledge-Based Systems*, Elsevier, v. 37, p. 480–489, 2013.