



**UFOP**

Universidade Federal  
de Ouro Preto

**Universidade Federal de Ouro Preto  
Instituto de Ciências Exatas e Aplicadas  
Departamento de Computação e Sistemas**

**Estudo e Definição de Modelos de  
Computação Evolucionária para o  
Problema de Roteamento de Veículos  
com Múltiplos Depósitos**

**Wagner Linhares Marques**

**TRABALHO DE  
CONCLUSÃO DE CURSO**

ORIENTAÇÃO:

Fernando Bernardes de Oliveira

COORIENTAÇÃO:

Rafael Frederico Alexandre

**Julho, 2019**

**João Monlevade–MG**

**Wagner Linhares Marques**

**Estudo e Definição de Modelos de Computação  
Evolucionária para o Problema de Roteamento  
de Veículos com Múltiplos Depósitos**

Orientador: Fernando Bernardes de Oliveira

Coorientador: Rafael Frederico Alexandre

Monografia apresentada ao curso de Engenharia de Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

**Universidade Federal de Ouro Preto**

**João Monlevade**

**Julho de 2019**

# FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA

## Estudo e Definição de Modelos de Computação Evolucionária para o Problema de Roteamento de Veículos com Múltiplos Depósitos

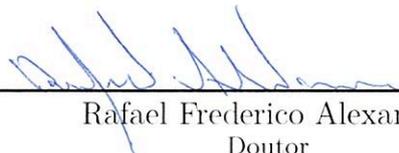
Wagner Linhares Marques

Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CSI496 – Trabalho de Conclusão de Curso II do curso de Bacharelado em Engenharia de Computação e aprovada pela Banca Examinadora abaixo assinada:



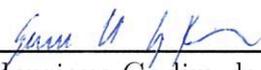
---

Fernando Bernardes de Oliveira  
Doutor  
DECSI – UFOP



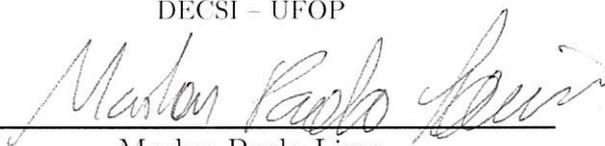
---

Rafael Frederico Alexandre  
Doutor  
DECSI – UFOP



---

George Henrique Godim da Fonseca  
Doutor  
Examinador  
DECSI – UFOP



---

Marlon Paolo Lima  
Doutor  
Examinador  
DECSI – UFOP

João Monlevade, 11 de julho de 2019

*Este trabalho é dedicado à minha família, quem incondicionalmente sempre esteve comigo na minha trajetória.*

# Agradecimentos

Agradeço primeiramente a Deus, por me proporcionar energia e persistência nos desafios da vida.

Aos meus pais e meu irmão, meu obrigado por me aturarem e não me deixar desistir dos meus sonhos.

Aos meus amigos, que sempre estiveram comigo e compartilharam muitas das etapas pelas quais tive que passar, muito obrigado. Em especial a Anderson, pela parceria e companheirismo durante todo esse tempo.

À Tainara, seu apoio foi essencial e sua energia me fortalece, muito obrigado.

E aos meus professores, em especial ao Fernando Bernardes e Rafael Alexandre, serei eternamente grato por todos ensinamentos nos quais vocês me proporcionaram. Vocês são ótimas pessoas e excelentes profissionais, que continuem sendo o exemplo que são.

À todos, muito obrigado!

*“Quem bater primeiro à dobra do mar. Dá, de lá, bandeira qualquer. Aponta pra fé e rema”*

—Los Hermanos,  
*em: Dois Barcos - 4.*

# Resumo

O Problema de Roteamento de Veículos com Múltiplos Depósitos (do inglês, *Multiple Deposit Vehicle Routing Problem* – MDVRP) é uma importante variante do clássico Problema de Roteamento de Veículos (PRV) (*Vehicle Routing Problem* – VRP). O MDVRP tem ampla aplicação prática e representa o contexto de diversas empresas. Como não existe até então um método capaz de obter a solução ótima do problema em tempo polinomial, este trabalho tem como objetivo o estudo do MDVRP a partir da construção de modelos de computação evolucionária. O problema é investigado no contexto mono-objetivo, considerando a redução do custo das rotas percorridas pelos veículos. Este trabalho apresenta uma meta-heurística híbrida baseada na Estratégia Evolutiva (*Evolution Strategy* – ES) com a incorporação de procedimentos de busca local. A estratégia aplicada é a  $(\mu + \lambda)$ -ES, a qual é responsável pela geração de descendentes a partir da mutação e a seleção dos indivíduos. A busca local é composta por nove estratégias, considerando os critérios de parada *Best Improvement* e *First Improvement*. A validação do algoritmo proposto foi realizada por meio de instâncias de teste disponíveis na literatura. Considerando o contexto experimental estabelecido, os resultados foram satisfatórios, sendo que alguns valores estão com um GAP de 10% em relação a literatura. Mesmo não atingindo o valor esperado em todas as instâncias, foi possível obter uma comparação com bons resultados e que podem ainda ser melhorados.

**Palavras-chaves:** Roteamento de Veículos. Problema de Roteamento de Veículos com Múltiplos Depósitos. Computação Evolucionária. Estratégias evolutivas.

# Abstract

The Multiple-Deposit Vehicle Routing Problem (MDVRP) is an important variant of the classic *Vehicle Routing Problem*. The MDVRP has broad practical application and represents the context of several companies. Since there is no computational method to obtain the optimal solution of the problem in polynomial time, this work has as objective the study of MDVRP from the construction of models of evolutionary computation. The problem is investigated in the mono-objective context, considering the cost minimization of vehicle routings. This work proposes a hybrid meta-heuristic based on the Evolutionary Strategy (*Evolution Strategy* – ES) with the incorporation of local search procedures. The  $(\mu + \lambda)$ -ES is used, which is responsible for the generation of offspring from the mutation and the selection of individuals. The local search is composed of nine strategies, considering as stop criteria the Best and the First Improvement. Computational experiments were used to evaluate the proposed method. Considering the established experimental environment, our method obtains GAP up to 10% from the literature. Even though the expected values for all instances were not obtained, the results suggest that our method provides competitive results and can also be improved.

**Key-words:** Vehicle Routing. The Multiple-Vehicle Vehicle Routing Problem. Evolutionary Computation. Evolutionary Strategy.

# Lista de ilustrações

Figura 1 – Diagrama dos <i>containers</i> e suas classes . . . . .	23
Figura 2 – Representação do Container de Depósito . . . . .	24
Figura 3 – Solução obtida pelo EsMDVRP para a P01 . . . . .	29
Figura 4 – Solução obtida pelo EsMDVRP para a P18 . . . . .	30
Figura 5 – Solução disponível na literatura para a P18 . . . . .	30
Figura 6 – Giant Tour por depósito da nova mutação . . . . .	31

# Lista de tabelas

Tabela 1 – Parâmetros do EsMDVRP . . . . .	27
Tabela 2 – Resultados médios do EsMDVRP . . . . .	27
Tabela 3 – Comparação do EsMDVRP com outros métodos – GAP (%) . . . . .	28
Tabela 4 – Comparação do EsMDVRP, valores médios, com outros métodos – GAP (%) . . . . .	29
Tabela 5 – Comparação do EsMDVRP: nova mutação com outros métodos – GAP (%) . . . . .	32

# Lista de abreviaturas e siglas

**AE** Algoritmos Evolutivos

**AG** Algoritmo Genético

**CLONALG** *Clonal Selection Algorithm*

**DE** *Differential Evolution*

**ES** *Evolution Strategy*

**MDVRP** *Multiple Deposit Vehicle Routing Problem*

**PSO** *Particle Swarm Optimization*

**PCV** Problema do Caixeiro Viajante

**PRVC** Problema de Roteamento de Veículos Capacitados

**PRV** Problema de Roteamento de Veículos

**VRP** *Vehicle Routing Problem*

# Lista de símbolos

$A$	Arcos
$C$	Capacidade dos veículos
$G = (V,A)$	Grafo completo
$K$	Veículos
$\lambda$	Número de descendentes.
$\mu$	Tamanho da população.
$V$	Vértices
$V_c$	Conjunto de clientes a serem atendidos
$V_D$	Conjunto de Depósitos

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	Objetivos	14
1.2	Metodologia	14
1.3	Organização do trabalho	15
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>16</b>
2.1	Formulação	16
2.2	Trabalhos correlatos	17
2.3	Considerações finais	18
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>19</b>
3.1	Estratégia Evolutiva	19
3.2	Método proposto para o MDVRP	20
3.3	Busca Local	22
3.4	Implementação	23
3.5	Considerações finais	25
<b>4</b>	<b>RESULTADOS</b>	<b>26</b>
4.1	Definição do experimento	26
4.1.1	Comparação dos resultados com outros métodos	27
4.1.2	Novo processo de mutação	30
4.2	Considerações Finais	32
<b>5</b>	<b>CONCLUSÃO</b>	<b>33</b>
	<b>REFERÊNCIAS</b>	<b>34</b>

# 1 Introdução

Os problemas de transportes envolvem diversos custos e questões a serem resolvidas. É preciso atender os clientes de modo hábil e eficiente, com custos aceitáveis. Uma das questões pertinentes a esse contexto é o Problema de Roteamento de Veículos (**PRV**), do inglês *Vehicle Routing Problem* (**VRP**). Este problema foi primeiramente estudado por meio do roteamento de veículos de distribuição de gasolina a postos de combustíveis, o qual foi generalizado do Problema do Caixeiro Viajante (**PCV**) (**DANTZIG; RAMSER, 1959**). A partir disso, diversos trabalhos são apresentados na literatura, com as mais diversas técnicas e métodos. O PRV é um problema combinatorial da classe NP-Difícil e até então não possui uma solução exata e ótima em tempo de processamento computacional polinomial (**OLIVEIRA et al., 2016**).

De modo geral, os problemas de roteamento consistem em definir rotas de atendimento a clientes, nas quais eles serão atendidos de maneira que atenda os requisitos. Uma quantidade de clientes deve ser atendida por um depósito. O depósito tem uma frota de veículos que pode ser homogênea ou não, e podem existir restrições acerca do tamanho da rota e da capacidade dos veículos. As rotas de atendimento, que são os percursos que os veículos vão seguir são definidas considerando que cada cliente é visitado uma única vez. Além disso, a soma das demandas dos clientes não pode exceder a capacidade de um veículo. Quando apenas a restrição de capacidade é definida, na literatura, ele é relacionado como Problema de Roteamento de Veículos Capacitados (**PRVC**) (**CORDEAU et al., 2007**). Os objetivos a serem considerados no **PRV** podem ser o tempo de atendimento, a menor quantidade de veículos, o menor percurso total, dentre outros.

O problema de roteamento de veículos com múltiplos depósitos (*Multi-Depot Vehicle Routing Problem* - **MDVRP**) é uma variante do **PRV** e consiste em incluir mais depósitos ao contexto para atender os clientes. O problema é mais complexo, sendo necessário definir por qual depósito o cliente será atendido, bem como em qual veículo (rota) ele será alocado. Assim, diversas rotas devem ser definidas, pois como tem mais depósitos, torna-se um problema com mais opções. Cada depósito possui uma frota limitada de veículos, a qual pode ser homogênea ou heterogênea. O **MDVRP** e suas variantes. Apesar do alto contexto prático, o **MDVRP** e suas variantes são pouco estudados na literatura, se comparados às outras classes do **PRV** (**CREVIER; CORDEAU; LAPORTE, 2007; MIRABI; GHOMI; JOLAI, 2010**).

O **MDVRP** pode ser representado pelo contexto de uma empresa com diversos pontos de distribuição (depósitos). A empresa precisa decidir de maneira adequada por qual depósito o cliente será atendido e principalmente, no menor tempo possível. Além

disso, outras questões podem ser utilizadas na abordagem, como a redução do número de veículos disponíveis na frota, a menor distância percorrida, dentre outros.

Para poder lidar melhor com os problemas do **MDVRP**, os Algoritmos Evolutivos (**AE**) são utilizados, eles são meta-heurísticas inspiradas na seleção natural e são estratégias eficientes para a resolução problemas de otimização. Em problemas NP-difíceis, **AE** podem obter soluções eficientes e competitivas em tempo aceitável (**NEDJAH; MOURELLE; ALBA, 2006**). Exemplos de algoritmos pertencentes a essa categoria são o Algoritmo Genético (**AG**), o *Clonal Selection Algorithm* (**CLONALG**), o *Differential Evolution* (**DE**), *Particle Swarm Optimization* (**PSO**), Colônia de Formigas, **ES**, dentre outros. Considerando o contexto do **MDVRP**, uma meta-heurística baseada em algoritmos evolutivos é proposto para o problema, utilizando o **ES**. Procedimentos de busca local foram incorporados ao algoritmo para o refinamento das soluções. Com isso, espera-se que o método proposto seja capaz de gerar soluções eficientes para o **MDVRP** e em tempo hábil.

## 1.1 Objetivos

O objetivo geral deste trabalho é o estudo do Problema de Roteamento de Veículos com Múltiplos Depósitos e a definição de um método baseado em técnicas de computação evolucionária para a geração de soluções eficientes e em tempo hábil para o problema.

Este trabalho possui os seguintes objetivos específicos:

1. Definir e implementar um algoritmo baseado em computação evolucionária para a geração de soluções eficientes para o **MDVRP**.
2. Estudar e incorporar mecanismos de busca local para o problema.
3. Validar o algoritmo proposto por meio de experimentos computacionais utilizando instâncias de teste disponíveis na literatura.

## 1.2 Metodologia

Os passos para a execução deste trabalho são definidos a seguir:

1. Revisar a literatura sobre o **MDVRP** acerca de trabalhos correlatos e demais conceitos, bem como identificar métodos e técnicas utilizadas na solução desse tipo de problema;
2. Estudar métodos para representação do problema, identificando abordagens e estruturas de dados específicas para o problema.
3. Definir e implementar uma meta-heurística para o problema baseada em técnicas de computação evolucionária;

4. Estudar e incorporar mecanismos de busca local aplicados ao problema;
5. Planejar e realizar experimentos para avaliar a performance da meta-heurística proposta;
6. Analisar e discutir os resultados obtidos, além de identificar possíveis melhorias e considerações gerais sobre o processo.

### 1.3 Organização do trabalho

O restante do trabalho é organizado como segue. A revisão bibliográfica, considerando os trabalhos correlatos na literatura e as respectivas abordagens para o problema, são apresentadas no Capítulo 2. No Capítulo 3 são descritas as estratégias a serem realizadas e como a modelagem é feita. O Capítulo 4 discute os resultados obtidos e a comparação com a literatura. Por fim, o Capítulo 5 apresenta as conclusões para o trabalho elaborado, além de apresentar propostas de melhorias futuras.

## 2 Revisão bibliográfica

Este capítulo apresenta inicialmente a formulação para o MDVRP na Seção 2.1, com os seus devidos detalhes. Em seguida, trabalhos correlatos na literatura são apresentados na Seção 2.2.

### 2.1 Formulação

O MDVRP pode ser formulado da seguinte maneira, como apresentado por Montoya-Torres et al. (2015), Vidal et al. (2012) e Oliveira et al. (2016). Seja  $G = (V, A)$  um grafo completo, no qual  $V$  são os vértices e  $A$  seus arcos. Os nós são divididos em dois subconjuntos: os clientes a serem atendidos  $V_C = \{1, \dots, N\}$ , e os depósitos  $V_D = \{N + 1, \dots, N + M\}$ , como  $V_C \cup V_D = V$  e  $V_C \cap V_D = \emptyset$ . Temos um custo não negativo  $c_{ij}$  associado a cada arco  $(i, j) \in A$ . A demanda de cada cliente é  $d_i$ . Existe também uma frota de  $K$  veículos, cada um com capacidade  $C$ .

Na formulação matemática apresentada a seguir, as variáveis binárias  $x_{ijk}$  são iguais a 1 quando o veículo  $k$  visita o nó  $j$  imediatamente depois do nó  $i$ . Variáveis auxiliares  $y_i$  são também usadas para eliminar as restrições.

$$\text{Min} \sum_{i=1}^{N+M} \sum_{j=1}^{N+M} \sum_{k=1}^K c_{ij} x_{ijk}, \quad (2.1)$$

$$\text{s.a. :} \sum_{i=1}^{N+M} \sum_{k=1}^K x_{ijk} = 1 (j = 1, \dots, N); \quad (2.2)$$

$$\sum_{j=1}^{N+M} \sum_{k=1}^K x_{ijk} = 1 (i = 1, \dots, N); \quad (2.3)$$

$$\sum_{i=1}^{N+M} x_{ihk} - \sum_{j=1}^{N+M} x_{hjk} = 0 (k = 1, \dots, K; h = 1, \dots, N + M) \quad (2.4)$$

$$\sum_{i=1}^{N+M} \sum_{j=1}^{N+M} d_i x_{ijk} \leq C (k = 1, \dots, K); \quad (2.5)$$

$$\sum_{i=N+1}^{N+M} \sum_{j=1}^{N+M} x_{ijk} \leq 1 (k = 1, \dots, K); \quad (2.6)$$

$$\sum_{i=1}^{N+M} \sum_{j=1+1}^{N+M} x_{ijk} \leq 1 (k = 1, \dots, K); \quad (2.7)$$

$$y_i - y_j + (M + N)x_{ijk} \leq N + M - 1; \text{ para } 1 \leq i \neq j \leq N \text{ e } 1 \leq k \leq K; \quad (2.8)$$

$$x_{ijk} \in \{0, 1\} \forall i, j, k. \quad (2.9)$$

$$y_i \in \{0, 1\} \forall i. \quad (2.10)$$

Em (2.1) o objetivo é minimizar o custo total. As restrições (2.2) e (2.3) garantem que cada cliente é atendido por exatamente um veículo. A conservação de fluxo é garantida por meio da restrição (2.4). A capacidade do veículo é encontrada em (2.5). As restrições (2.6) e (2.7) verificam a disponibilidade do veículo. Em (2.8) são as restrições de eliminação de sub-rotas. Por fim, em (2.9) e (2.10) definem  $x$  e  $y$  como as variáveis binárias (OLIVEIRA et al., 2016).

## 2.2 Trabalhos correlatos

O estudo do MDVRP é relevante em virtude do alto contexto prático. Ele é condizente com a realidade de diversas empresas nos mais diversos segmentos de mercado. Apesar disso, o número de estudos na literatura sobre o MDVRP é, em geral, limitado, quando comparado ao VRP. Um levantamento desses estudos, baseados em métodos exatos ou heurísticas, é apresentado por Montoya-Torres et al. (2015). Métodos inspirados em modelos evolutivos são encontrados na literatura. O AG é uma técnica clássica nesse contexto, e aplicações ao MDVRP foram realizadas. Um estudo dos diferentes tipos de AG aplicados ao MDVRP podem ser acessados em Karakatič e Podgorelec (2015). Métodos aplicados ao MDVRP e suas variantes podem ser obtidos em Gendreau et al. (2008).

Embora diferentes métodos tenham sido aplicados ao VRP, um número restrito de trabalhos foi definido para o MDVRP considerando a minimização do custo das rotas (MONTROYA-TORRES et al., 2015; RAY et al., 2014; RAHIMI-VAHED et al., 2015; KUO; WANG, 2012; SALHI; IMRAN; WASSAN, 2014; CONTARDO; MARTINELLI, 2014).

Vidal et al. (2012) propuseram um modelo evolucionário híbrido, composto por vários operadores de busca local. Sua meta-heurística combina o contexto de Algoritmo Genético com as buscas por vizinhança, propondo um algoritmo chamado *Hybrid Genetic Search with Adaptive Diversity Control*. Os autores dividiram sua população em dois subconjuntos (subpopulações), sendo um deles com as soluções factíveis e outro com soluções infactíveis, aplicando penalidades aos excessos de cargas e de tempo de viagem. Além disso, eles utilizaram processos definidos como “Educação” e “Reparo” nas subpopulações para tentar torná-las factíveis. Na realidade, a “Educação” é um processo de busca local. Sua representação foi feita baseada em um *Giant Tour*, no qual ele armazenava as rotas, distribuídas em depósitos e períodos em um único cromossomo, que representava a sequência dos clientes. Ao fim do processo aplicado, o método realizava a seleção dos indivíduos para cruzamento por meio de torneios binários da população, no qual poderiam estar incluídas,

tanto as soluções factíveis ou não. Após isso, ainda foi aplicado um operador de cruzamento para combinar as características dos cromossomos-pais, o que foi denominado por *Periodic Crossover with Insertions (PIX)*. Esse algoritmo possibilitou que eles obtivessem soluções melhores, de maneira que a atuação de todo conjunto proposto levou o algoritmo a alcançar resultados satisfatórios.

Um modelo coevolutivo cooperativo foi definido em [Oliveira et al. \(2016\)](#), considerando um ambiente paralelo para resolver o [MDVRP](#). Os resultados obtidos sugerem que o algoritmo produz soluções competitivas, quando são comparadas com as melhores soluções conhecidas. Além disso, considerando o contexto experimental, os resultados sugerem que o método pode ser mais rápido quando comparado com outras técnicas da literatura.

## 2.3 Considerações finais

Diferentes abordagens foram propostas para a solução do [MDVRP](#). Os trabalhos apresentados foram fundamentais para o entendimento do problema e das estratégias, bem como auxiliaram no desenvolvimento do trabalho proposto. No [Capítulo 3](#) é apresentada a técnica utilizada como base, bem como as discussões acerca da implementação.

## 3 Desenvolvimento

Este capítulo contextualiza a técnica utilizada como base para definição do algoritmo. Em seguida, a implementação do método é discutida, considerando os principais componentes e como as informações são gerenciadas.

### 3.1 Estratégia Evolutiva

Para a abordagem do [MDVRP](#), foi proposta uma Estratégia Evolutiva, que é uma técnica de otimização que baseia-se nos conceitos de adaptação e evolução. Essa estratégia evolutiva, denominada [ES](#), pertence à classe de algoritmos de Computação Evolucionária, no qual dado uma população com  $\mu$  indivíduos, são gerados  $(\lambda / \mu)$  descendentes que foram modificados pela mutação. Dessa população, apenas os  $\mu$  melhores indivíduos permanecem para próxima geração.

O ES caracteriza-se por ser relativamente rápido e possuir uma aplicabilidade satisfatória para valores reais. A partir de suas futuras gerações, o algoritmo avalia se a geração corrente é melhor que a anterior, considerando sempre a melhora. O ES usa um critério para seleção de sobreviventes, podendo ser  $(\mu, \lambda)$ -ES ou  $(\mu + \lambda)$ -ES ([BÄCK, 1996](#)).

A partir de  $\mu$  indivíduos (tamanho da população inicial), obtêm-se  $\lambda / \mu$  descendentes. Na abordagem  $(\mu, \lambda)$ -ES, os descendentes substituem os  $\mu$  pais para a próxima geração. O contexto  $(\mu + \lambda)$ -ES indica que haverá concorrência entre os  $\mu$  indivíduos com os descendentes para a próxima geração. Os indivíduos, pais e descendentes, são combinados em uma única população, classificados e ordenados. Uma nova população de tamanho  $\mu$  é selecionada. Os demais indivíduos são descartados. O Algoritmo 1 mostra o modelo canônico  $(\mu + \lambda)$ -ES, o qual foi retirado de [Luke \(2013\)](#).

**Algoritmo 1** Estratégia Evolutiva –  $(\mu + \lambda)$ 


---

```

1:  $\mu \leftarrow$  número de pais selecionados
2:  $\lambda \leftarrow$  número de filhos gerados pelos pais
3:  $P \leftarrow \{ \}$  ▷ Conjunto de soluções
4: for  $\lambda$  do
5:    $P \leftarrow P \cup$  novo indivíduo aleatório
6:  $Best \leftarrow \emptyset$ 
7: for cada indivíduo  $P_i \in P$  do
8:    $AvaliarCusto(P_i)$ 
9:   if  $Best = \emptyset$  ou  $AvaliarCusto(P_i) > AvaliarCusto(Best)$  then
10:     $Best \leftarrow P_i$ 
11:  $Q \leftarrow$  os  $\mu$  indivíduos em  $P$  que possuem o melhor custo
12:  $P \leftarrow \{ \}$ 
13: for cada indivíduo  $Q_j \in Q$  do
14:   for  $\lambda/\mu$  descendentes do
15:     $P \leftarrow P \cup \{Muta\c{c}{a}{o}(Copia(Q_j))\}$ 
return  $Best$  é a solução ideal ou a que temos quando o tempo acabou.

```

---

Na Seção 3.2, é definido o método proposto que utilizará uma derivação do Algoritmo 1 aplicando técnicas de Busca Local para que seja possível obter um refinamento maior dos resultados.

## 3.2 Método proposto para o MDVRP

Como o ES possui como operador principal a mutação e não depende de outros indivíduos para a geração de descendentes, ele foi utilizado como base para a abordagem proposta. A utilização da operação de *crossover*, como no Algoritmo Genético, poderia gerar soluções ineficazes para o MDVRP, dependendo da codificação e do operador de cruzamento. Ao algoritmo proposto são incorporados procedimentos de busca local, os quais permitem um refinamento das soluções. O modelo pode ser caracterizado como um modelo híbrido, que utiliza a Estratégia Evolutiva aliada às técnicas de busca local.

O método proposto, denominado *EsMDVRP*, é apresentado no Algoritmo 2. Os detalhes de cada parte do algoritmo são discutidos a seguir. Os parâmetros de entrada são: *Container* de Clientes, *Container* de Depósitos, tamanho da população ( $\mu$ ), quantidade de descendentes ( $\lambda$ ), número de gerações, e os parâmetros de probabilidade de mutação (*PBM*), probabilidade de busca local (*PBL*).

Nas linhas 2 e 3 são feitas as atribuições do número de pais selecionados ( $\mu$ ) e dos descendentes que serão gerados a partir dos pais ( $\lambda$ ). Na linha 4 é representada população. Na linha 5, a população inicial é gerada. Para isso, os clientes são alocados aos depósitos mais próximos. Na população inicial, os clientes são alocados aos depósitos mais próximos. Depois disso, são sorteados três clientes de maneira aleatória e o mais próximo a este

**Algoritmo 2** Estratégia Evolutiva –  $(\mu + \lambda)$ 


---

```

1: procedure EsMDVRP(PBM, PBL,  $\mu$ ,  $\lambda$ , numeroDeGeracoes)
2:    $\mu \leftarrow$  número de pais selecionados
3:    $\lambda \leftarrow$  número de filhos gerados pelos pais
4:    $S \leftarrow \{ \}$  ▷ Conjunto de soluções
5:    $i \leftarrow 1$ 
6:    $S \leftarrow$  GerarPopulacaoInicial( $\mu$ ) ▷ Gera a população inicial com os  $\mu$  indivíduos
7:   CalcularAptidao( $S$ ) ▷ Avalia cada indivíduo da população
8:   while (  $i <$  numeroDeGeracoes) do ▷ Enquanto número de gerações não for
   atendido
9:      $D \leftarrow \{ \}$  ▷ Conjunto de descendentes gerados
10:    for para cada solução pai do
11:       $k \leftarrow 1$ 
12:       $j \leftarrow 1$ 
13:      for ( $\lambda/\mu$ ) descendentes do
14:         $D_j \leftarrow$  Mutacao( $S_k$ , PBM) ▷ Muta o indivíduo  $j$ 
15:        CalcularAptidao( $D_j$ ) ▷ Calcula a aptidão do indivíduo  $j$ 
16:        if (  $rand(0.0, 1.0) <$  PBL) then ▷ Verifica se o valor aleatório é menor
   que o parâmetro, se sim, realiza Busca Local
17:           $D_j \leftarrow$  BuscaLocal( $D_j$ ) ▷ Nesse caso, a busca local é First
   Improvement
18:           $D_j \leftarrow$  CalcularAptidao( $D_j$ ) ▷ Avalia o novo indivíduo
19:          Adicionar( $S$ ,  $D$ ) ▷ Adicionar o descendente gerado ao conjunto D
20:           $j \leftarrow j+1$ 
21:           $k \leftarrow k+1$ 
22:           $S \leftarrow$  Selecionar( $S$ ,  $D$ ,  $\mu$ ) ▷ Selecione os  $\mu$  melhores indivíduos ( $\mu + \lambda$ ) de  $S + D$ 
   de acordo com o custo
23:           $S_0 \leftarrow$  MelhorIndividuo( $S$ ) ▷ Recupera o melhor indivíduo até o momento
24:          BuscaLocalBestImprovement( $S_0, S_1, S_{mut}$ ) ▷ Aplica a Busca Local Best
   Improvement aos dois melhores indivíduos e ao último mutado.
25:           $i \leftarrow i+1$ 
26:   return  $S_0$  ▷ Retorna a melhor solução em função de custo encontrada
Fim

```

---

é adicionado ao depósito. Após a geração da população, na linha 7 é feito o cálculo da aptidão de cada um dos indivíduos gerados, de acordo com a Equação 2.1.

Enquanto o número total de gerações não for atingido (linha 8), o algoritmo é executado. Na linha 9, o conjunto dos descendentes que serão gerados é iniciado. Na linha 13, para  $(\lambda/\mu)$  descendentes, os pais são selecionados de maneira aleatória na linha 14, para que a mutação ocorra ou não no indivíduo selecionado. O indivíduo é clonado, e a mutação consiste em aplicar uma perturbação a essa cópia. A perturbação consiste em permutar clientes (*swap*) do mesmo depósito ou de depósitos diferentes. Para que seja realizada a mutação, é aplicado a técnica de *swap* nos clientes, que verifica se a troca respeita as restrições, caso respeite, ela é realizada independente de melhora ou não na solução atual.

Na linha 16 é gerado um valor aleatório, para que a busca local ocorra ou não no indivíduo selecionado (linha 17). A busca local aplica as estratégias apresentadas a seguir na Seção 3.3. Após esses operadores, na linha 18 é calculada a aptidão do descendente gerado. Na linha 19 o indivíduo gerado é adicionado ao conjunto de descendentes. Após a geração de todos os descendentes, os  $\mu$  melhores indivíduos são selecionados a partir da união dos indivíduos da geração corrente e dos descendentes (linha 22).

O melhor indivíduo é recuperado na linha 23. Para um refinamento do melhor indivíduo da população, a busca local é aplicado com critério *Best Improvement* (linha 24). Além disso, a busca local é aplicada no segundo melhor indivíduo da população e no recém mutado. O contador de gerações é incrementado na linha 25. Quando o número de gerações é alcançado, o algoritmo retorna a solução que obteve o menor custo,  $S_0$  (linha 263).

Devido ao limite de veículos estabelecido pelo problema, o EsMDVRP penaliza o custo das soluções que possuem mais veículos do que o definido pela instância. Dessa maneira, isso faz com que os operadores procurem sempre diminuir o número de veículos, de forma a atender os requisitos do problema.

### 3.3 Busca Local

Procedimentos de busca local foram incorporados ao algoritmo para o refinamento das soluções. Eles são aplicados ao longo das gerações dado o parâmetro estabelecido. O procedimento foi definido por Prins (2004) e também utilizado por Vidal et al. (2012). Essa abordagem permite percorrer a vizinhança da solução de modo que busque por um valor melhor ao já obtido. Considerando que  $u$ ,  $x$ ,  $v$ ,  $y$  são clientes, e que  $x$  é o cliente após  $u$  ( $u \rightarrow x$ ) e  $y$  é o cliente após  $v$  ( $v \rightarrow y$ ), os movimentos são definidos como segue:

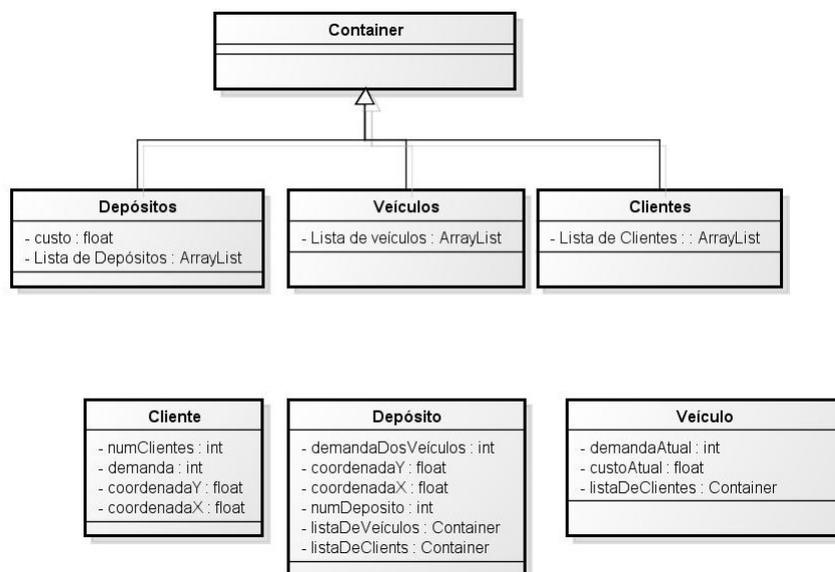
1. Mover um cliente  $u$  e realocá-lo após outro cliente  $v$ ;
2. Mover os clientes  $u$  e  $x$  para depois do cliente  $v$ , nessa ordem;
3. Mover os clientes  $u$  e  $x$  para depois do cliente  $v$ , em ordem inversa ( $x, u$ );
4. Trocar as posições dos clientes  $u$  e  $v$ ;
5. Trocar as posições dos clientes  $u$  e  $x$  pela posição de  $v$ ;
6. Trocar as posições dos clientes  $u$  e  $x$  pelas de  $v$  e  $y$ ;
7. Se  $u$  e  $v$  pertencem à mesma rota, trocar os arcos  $(u, x)$  e  $(v, y)$  por  $(u, v)$  e  $(x, y)$ ;
8. Se  $u$  e  $v$  pertencem a rotas distintas; trocar os arcos  $(u, x)$  e  $(v, y)$  por  $(u, v)$  e  $(x, y)$ ;
9. Se  $u$  e  $v$  pertencem a rotas distintas; trocar os arcos  $(u, x)$  e  $(v, y)$  por  $(u, y)$  e  $(x, v)$ .

Essa estratégia tem duas maneiras de ser aplicada. A primeira utiliza o critério *First Improvement* e a segunda o *Best Improvement*. Na primeira, o refinamento é interrompido na primeira melhora em relação ao custo atual. Na segunda maneira, possui um custo computacional muito grande, dado que a vizinhança é percorrida por completo para que possa obter o melhor resultado (melhor vizinho), devido a isso ela é aplicada com uso de *threads*. No trabalho proposto é aplicado a *First Improvement* aos  $(\lambda/\mu)$  descendentes e ao fim do processo, a *Best Improvement* é aplicada ao final de cada geração, desde que tenha *threads* livres.

### 3.4 Implementação

O modelo proposto foi implementado na linguagem JAVA com a IDE NetBeans. Para representar a solução e os demais componentes do problema, foram utilizados *containers*, de modo que possa conter as informações necessárias para cada um dos dados relacionados. Um *container* é um conjunto dos principais tipos com suas informações associadas. Por exemplo, se for um *Container* de Clientes, ele possui todas as informações relacionadas a cada cliente associado. Esse modo de representação possibilita generalizar em um nível maior cada um dos itens, permitindo uma manipulação mais fácil do problema. Uma representação dos *containers* é mostrada na Figura 1.

Figura 1 – Diagrama dos *containers* e suas classes



Fonte: elaborado pelo autor

No modelo proposto são utilizados três tipos de *containers*: veículos, clientes e depósitos. Um *container* de Clientes possui uma lista de Clientes para poder armazená-los. Semelhante ao anterior, o *container* de Veículos, possui uma lista de veículos associados a

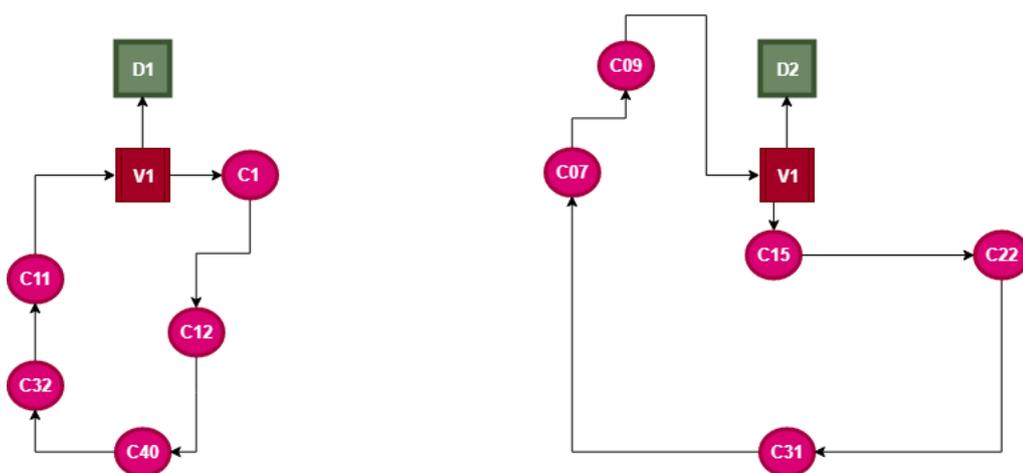
ele. O principal *container* é o *container* de Depósitos, pois ele representa a solução principal do problema. Uma população é composta por uma lista de *containers* de Depósitos, pois ele possui todas informações necessárias para análise, como o custo total, custo de cada um dos veículos com suas respectivas rotas e clientes associados.

As três principais classes são: Cliente, Depósito e Veículo. O Cliente é composto por sua demanda em ser atendido, suas coordenadas e o seu número correspondente. O Veículo possui um custo associado, que seria o custo de sua rota definida, a demanda máxima que ele suporta, demanda atual e os Clientes que devem ser atendidos por ele. Um depósito possui coordenadas associadas, número do depósito e uma lista de clientes e veículos alocados a ele.

Durante a implementação da *Best Improvement* foi notado uma necessidade de que ela ocorra de modo paralelo, dado o seu alto custo de processamento. Dessa maneira, são utilizados três *threads* para executar as rotinas durante as gerações.

Uma solução para o MDVRP é representada por um *Container* de Depósito, como mostrado na Figura 2. Esse *container* é composto por um *container* de Veículo no qual cada veículo possui clientes associados e uma rota de visitação. O número de veículos é variável em relação à distribuição de clientes a cada depósito, de modo que só tenha aqueles que possuem clientes alocados. A função de avaliação, para obter o custo é baseado na distância euclidiana de cliente para cliente e cliente para depósito, sendo o custo total de cada veículo o somatório do custo da rota e o custo da solução a soma de todos os custos. O custo é calculado de acordo com a Equação 1 definida na Seção 2.1. A Figura 2 mostra como é feita a representação do *container* de depósitos com uma rota definida.

Figura 2 – Representação do Container de Depósito



Fonte: elaborado pelo autor

## 3.5 Considerações finais

Após finalizar a modelagem e definir os métodos de aplicação, os testes foram feitos para que seja encontrado parâmetros mais apropriados ao problema. No próximo capítulo são apresentados os valores obtidos e uma discussão dos resultados que foram gerados para cada uma das instâncias de teste.

## 4 Resultados

Este capítulo apresenta o contexto experimental estabelecido para a validação do algoritmo proposto. Em seguida, os resultados são analisados e discutidos.

### 4.1 Definição do experimento

Para validação do método proposto foram utilizadas instâncias de teste disponíveis na literatura. As instâncias foram definidas por [Cordeau et al. \(2007\)](#), e foram utilizadas por [Vidal et al. \(2012\)](#) e [Oliveira et al. \(2016\)](#)<sup>1</sup>. Considerando o algoritmo proposto, foram selecionadas 11 das 33 instâncias de teste disponíveis. As demais não foram utilizadas pois possuem restrição de custo de rota e o algoritmo proposto não conseguiu gerar soluções factíveis para instâncias com esse tipo de restrição.

Durante a análise empírica do experimento, foi possível observar que o algoritmo em algumas situações retornava soluções infactíveis, devido ao fato de possuir mais veículos do que a quantidade definida pela instância. Dessa maneira, foi adicionado ao cálculo de função objetivo uma penalização para cada veículo excedente. Como penalidade, foi somando ao custo total uma multa de 1.000 unidades por cada veículo excedente. Essa estratégia contribui para a geração de soluções factíveis, pois com a minimização do custo, o algoritmo tende a reduzir o número de veículos até o limite permitido.

Os parâmetros utilizados pelo algoritmo foram definidos de maneira empírica a partir de testes preliminares. O limite de gerações foi determinado a partir de um valor médio que atendesse maioria das instâncias. Assim, os seguintes parâmetros foram estabelecidos: ( $\mu = 5$ ), ( $\lambda = 10$ ), 3.000 gerações, taxa de mutação igual a 0,4 e taxa de busca local igual a 0,9. As instâncias de teste selecionadas, com suas respectivas características, estão disponíveis na Tabela 1.

O algoritmo foi executado por 30 vezes para cada uma das instâncias, sendo que as observações foram aleatorizadas. Todas as execuções foram realizadas num computador com processador Intel Core i3, 4GB de memória RAM e sistema operacional *Windows 7*.

O critério de parada utilizado é o número de gerações. Se durante o encerramento das gerações, alguma *thread* ainda está executando sobre uma solução, o EsMDVRP aguarda seu término e realiza o ranqueamento de soluções somente após o término de todas execuções. Quando o critério é atingido, a execução para. Se a melhor solução até aquele momento torna-se a solução final.

<sup>1</sup> As instâncias estão disponíveis em: <<https://github.com/fboliveira/MDVRP-Instances>>

Tabela 1 – Parâmetros do EsMDVRP

Num	Instancia	Clientes	Depósitos	Veículos
1	P01	50	4	50
2	P02	50	4	50
3	P03	75	5	75
4	P04	100	2	100
5	P05	100	2	100
6	P06	100	3	100
7	P07	100	4	100
8	P12	80	2	80
9	P15	160	4	160
10	P18	240	6	240
11	P21	360	9	360

Fonte: dados da pesquisa

Os resultados para o EsMDVRP, considerando o tempo médio de execução (em minutos), o valor de custo médio, o melhor custo encontrado são apresentados na Tabela 2.

Tabela 2 – Resultados médios do EsMDVRP

Num	Instancia	Custo médio	Tempo (min)	Melhor custo
1	P01	607,22	4,00	<b>576,87</b>
2	P02	501,02	15,76	480,04
3	P03	687,39	10,38	665,99
4	P04	1.094,98	12,04	1.058,26
5	P05	838,31	43,53	776,42
6	P06	957,98	11,15	917,36
7	P07	968,29	11,33	937,52
8	P12	1.444,45	28,13	1365,3
9	P15	2.953,73	40,88	2.765,11
10	P18	4.595,72	52,21	4.243,48
11	P21	7.599,40	69,81	7.031,52

Fonte: dados da pesquisa

#### 4.1.1 Comparação dos resultados com outros métodos

Os resultados do EsMDVRP são comparados com outros métodos da literatura, considerando a média dos custos e o melhor custo para cada uma das instâncias utilizadas. Esses resultados são comparados por meio do *GAP*, o qual é calculado por meio da Equação 4.1. Naquela equação, EsMDVRP se refere aos valores obtidos pelo método proposto, enquanto *VL* representa o valor do método da literatura em questão.

$$GAP(\%) = \frac{EsMDVRP - VL}{VL} \times 100 \quad (4.1)$$

A Tabela 3 apresenta a comparação do EsMDVRP com dois métodos. O primeiro deles (CGL97) foi proposto por [Cordeau, Gendreau e Laporte \(1997\)](#) e o segundo (HGSADC) foi proposto por [Vidal et al. \(2012\)](#). Como é reportado para o HGSADC o melhor resultado obtido por aqueles autores, ele foi comparado com o melhor resultado do EsMDVRP.

Tabela 3 – Comparação do EsMDVRP com outros métodos – GAP (%)

Num	Instância	CGL97	HGSADC	HGSADC_Melhor
1	P01	0,00	0,00	0,00
2	P02	1,37	1,37	1,37
3	P03	3,23	3,96	3,96
4	P04	5,13	5,76	5,91
5	P05	3,06	3,52	3,52
6	P06	4,50	4,66	4,66
7	P07	5,11	6,30	6,30
8	P12	3,51	3,51	3,51
9	P15	9,11	10,37	10,37
10	P18	14,36	14,60	14,60
11	P21	27,01	28,43	28,43

Fonte: dados da pesquisa

Considerando o contexto experimental estabelecido, os resultados sugerem que, das 11 instâncias utilizadas, 5 possuem um GAP menor que 5%; 4 estão entre 5% e 10%, 2 estão entre 15% e 29%. As instâncias mais complexas, como a P18, P21 o GAP obtido foi o maior dentre as demais. Isso é devido a complexidade das instâncias e a maneira que o algoritmo constrói suas soluções.

Como foram realizadas 30 execuções de maneira aleatória e considerando o contexto estocástico do método, uma análise dos valores médios obtidos em relação a literatura também é apresentada. Na Tabela 4 é possível observar os resultados obtidos.

Para o valor médio obtido pelo algoritmo, das 11 instâncias utilizadas, 7 obtiveram um GAP inferior a 10%, 1 instância obteve um gap entre 10 - 15% e 3 instâncias obtiveram um gap acima de 15% - 39%. Esses resultados sugerem que em média, o GAP para instâncias foi baixo, sendo inferior a 15% em sua maioria. Isso sugere que o EsMDVRP é capaz de gerar soluções respeitando os critérios do problema.

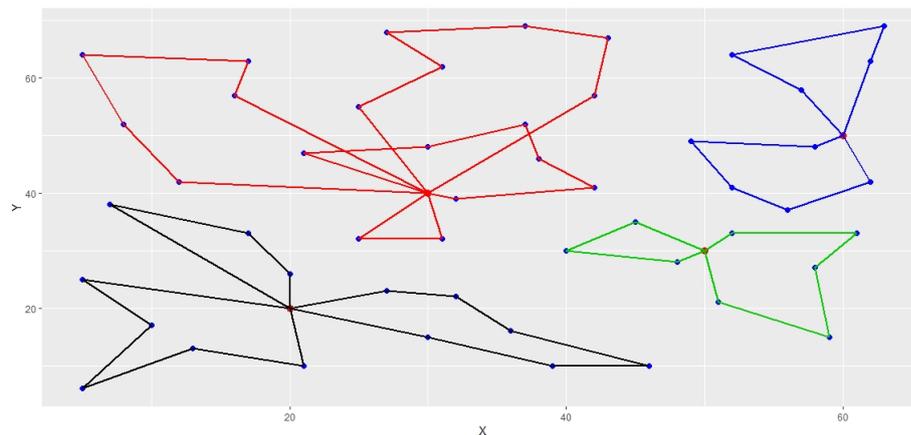
Na Figura 3 é possível observar a solução obtida para P01 no melhor caso. Para essa instância, o algoritmo conseguiu alcançar o melhor resultado, considerando que esse é o mínimo valor possível provado pela literatura ([VIDAL et al., 2012](#)).

Tabela 4 – Comparação do EsMDVRP, valores médios, com outros métodos – GAP (%)

Num	Instância	CGL97	HGSADC	HGSADC_Melhor
1	P01	5,26	5,26	5,26
2	P02	5,73	5,81	5,81
3	P03	6,55	7,3	7,3
4	P04	8,77	9,43	9,58
5	P05	11,28	11,77	11,77
6	P06	9,3	9,3	9,3
7	P07	8,56	9,79	9,79
12	P12	9,52	9,52	9,52
15	P15	16,56	17,89	17,89
18	P18	23,86	24,11	24,11
21	P21	37,27	38,81	38,81

Fonte: dados da pesquisa

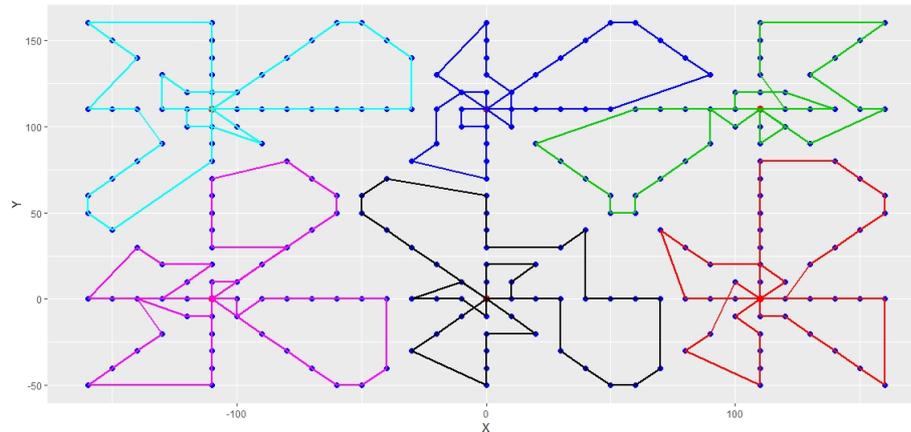
Figura 3 – Solução obtida pelo EsMDVRP para a P01



Fonte: elaborado pelo autor

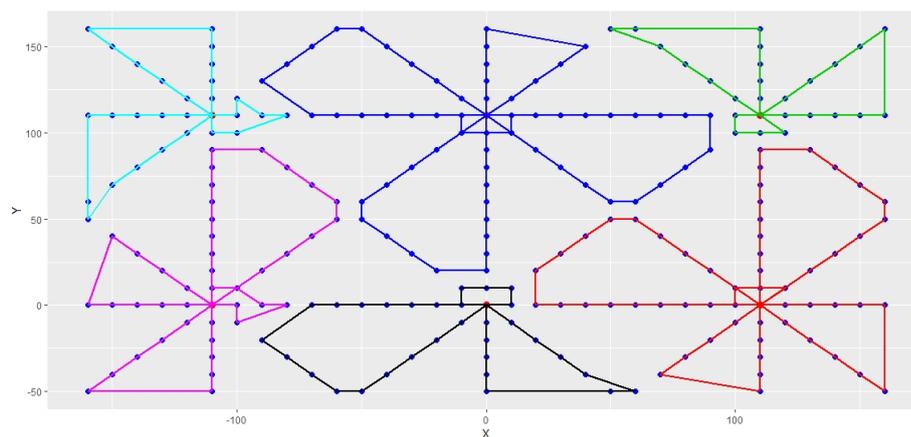
A Figura 4 mostra a solução obtida para a instância P18. Essa solução possui um GAP de aproximadamente 14,60%. A melhor solução disponível na literatura é mostrada na Figura 5. É possível notar semelhanças entre as soluções. A solução disponível pela literatura possui uniformidade entre suas rotas, enquanto a do EsMDVRP ainda precisa de um melhor refinamento.

Figura 4 – Solução obtida pelo EsMDVRP para a P18



Fonte: elaborado pelo autor

Figura 5 – Solução disponível na literatura para a P18



Fonte: elaborado pelo autor

Mesmo o custo da solução não ter sido igual ao da literatura, a solução obtida na Figura 4 é factível. Ela ser factível e respeitar as restrições da instância é um fator muito relevante, dado sua complexabilidade. No geral, um aperfeiçoamento das técnicas de busca local ou uma nova representação podem trazer melhorias nos resultados obtidos e serão avaliados em trabalhos futuros.

#### 4.1.2 Novo processo de mutação

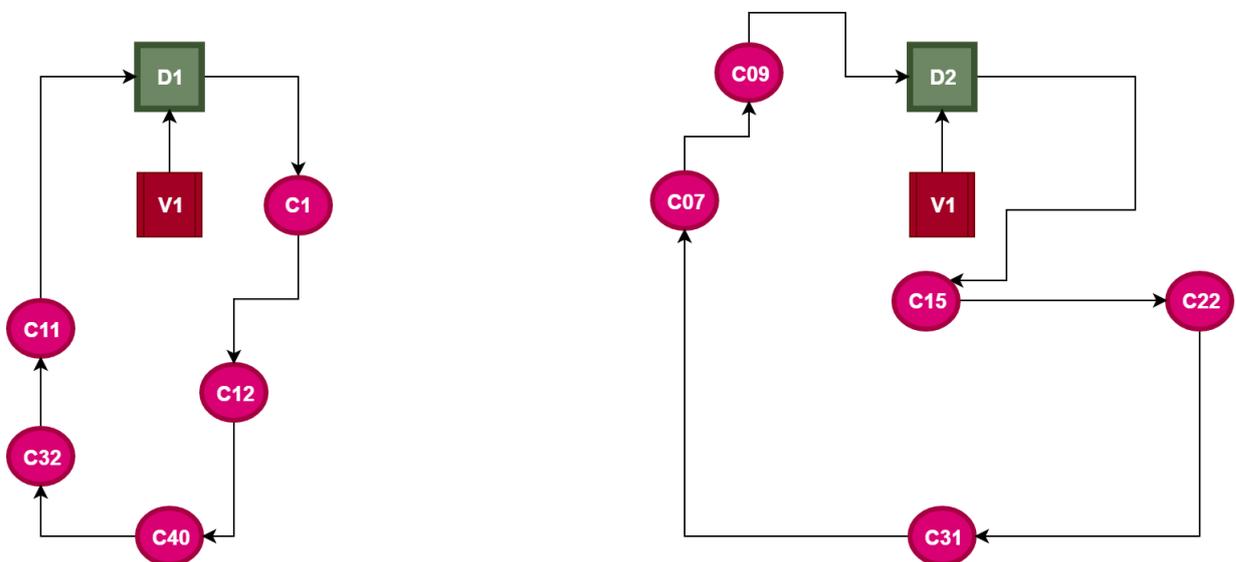
Durante a realização de experimentos, observou-se que a mutação poderia limitar a variabilidade da solução e diminuir a possibilidade de ter um custo menor. A primeira mutação realizada aplicava o procedimento de *swap* na solução, respeitando as restrições

tal como a busca local. A diferença entre ela e a busca local consiste que, se a troca não melhorasse o custo da solução, ela seria feita da mesma maneira.

Analisando o comportamento do algoritmo, em algumas situações as trocas não eram feitas nas rotas pois alguma restrição poderia inviabilizar a operação, como a demanda, por exemplo. Devido a isso, foi implementado uma nova mutação, denominada *New Mutation*, que consiste gerar listas de clientes relacionadas aos depósitos e, com isso, realizar o *swap* de uma maneira independente e sem restrições. Esse processo é similar à criação de *giant tour*, como definido em Prins (2004).

Como a solução é composta por *containers* de depósitos, cada depósito do *container* possui uma frota de veículos associada a ele e essa frota possui uma lista de cliente com rotas que são geradas pelo EsMDVRP. Para que o *giant tour* seja realizado, a estratégia adotada foi retirar todos clientes dos veículos e deixá-los apenas alocados aos depósitos. Desse modo, os *swaps* podem ser feitos independentemente da demanda ou custo da rota do veículo. Terminado o *swap*, a distribuição dos clientes é realizada de maneira gulosa pela demanda, respeitando as restrições do problema. Caso seja necessário um novo veículo, ele é adicionado ao *container*. Na figura 6 é possível ver como fica a representação em *Giant Tour* do processo.

Figura 6 – Giant Tour por depósito da nova mutação



Fonte: elaborado pelo autor

A Tabela 5 mostra o GAP para os melhores resultados obtidos em comparação à literatura, com os mesmos parâmetros da rotina anterior, definidos na Seção 4.1. A única modificação é em relação à taxa de mutação, que neste experimento foi de 10%.

Comparando os resultados da Tabela 3 e da Tabela 5 é possível notar que os valores

Tabela 5 – Comparação do EsMDVRP: nova mutação com outros métodos – GAP (%)

Num	Instância	CGL97	HGSADC	HGSADC_Melhor
1	P01	4,98	4,98	4,98
2	P02	1,86	1,93	1,93
3	P03	2,26	2,97	2,97
4	P04	20,15	20,87	21,04
5	P05	4,02	4,48	4,48
6	P06	9,68	9,85	9,85
7	P07	3,30	4,47	4,47
8	P12	4,57	4,57	4,57
9	P15	11,23	12,50	12,50
10	P18	20,29	20,54	20,54
11	P21	47,47	49,12	49,12

Fonte: dados da pesquisa

foram mais elevados em todas as instâncias de testes, com exceção da P07. Dado que o EsMDVRP possui uma penalização para gerar rotas com veículos a mais do que é definido na instância, isso faz com que ele gere soluções com custos muito elevado. Desse modo, existe uma necessidade maior do processo de Busca Local para reduzir esses veículos excedentes. Em contrapartida, a mutação anteriormente utilizada, não adicionava novos veículos, pois ela apenas tentava realizar trocas ignorando a melhora na solução ou não.

Como essa nova mutação utiliza um *giant tour* de clientes alocados ao Depósito, não é verificado se a troca gera ou não um novo veículo. Com isso, a perturbação que a mutação causa na população é maior, mesmo com parâmetros baixos, como utilizado neste experimento.

Essa característica causou um dano maior na penalização que o algoritmo atribui a cada veículo a mais. Desse modo, os resultados obtiveram um GAP superior ao da mutação original. O EsMDVRP acaba tendo que trabalhar para reduzir os veículos e consequentemente não consegue otimizar as soluções tal como a metodologia proposta anteriormente.

## 4.2 Considerações Finais

De maneira geral, os resultados sugerem que o algoritmo apresentou soluções que respeitam as restrições do problema. O desempenho do algoritmo pode ser considerado satisfatório, com suas ressalvas. Novas maneiras de tornar a execução mais rápida podem ser estudadas em trabalhos futuros, e espera-se que sejam obtidos valores melhores em relação aos da Tabela 2. No próximo capítulo, as conclusões serão discutidas e as propostas de continuidade do projeto serão apresentadas.

## 5 Conclusão

Este trabalho apresentou a definição de um modelo de computação evolucionária baseado no algoritmo de Estratégia Evolutiva para geração de soluções para Problema de Roteamento de Veículos com Múltiplos Depósitos. Este problema tem aplicação prática e o seu estudo pode auxiliar no processo de redução de custos. O modelo proposto é uma versão híbrida, com a incorporação de mecanismos de busca local.

A revisão bibliográfica foi importante para o entendimento do problema, bem como a identificação de abordagens correlatas para o MDVRP. As abordagens identificadas foram úteis para a proposta do algoritmo e dos mecanismos de busca local.

De modo geral, os resultados sugerem valores relativamente próximos aos da literatura. As estratégias de aplicação atenderam às restrições do problema e aos objetivos deste projeto. As soluções obtidas em algumas instâncias tiveram um GAP em torno de 10%. Entretanto, constatou-se que algumas das instâncias de testes não alcançaram resultados bons quando comparados aos reportados na literatura. Fatores como execução em paralelo ou uma abordagem diferente para a busca local devem ser investigadas.

Considerando os objetivos para o projeto, é possível identificar que foi estabelecida uma estratégia capaz de gerar soluções para o MDVRP. Essa abordagem é robusta e permite que modificações e outras ações sejam incorporadas ao modelo, para torná-lo mais rápido e mais eficiente.

Como propostas de continuidade do projeto foram identificados as seguintes ações principais:

- Aprimorar as técnicas de busca local, seja em implementação ou pela estratégia adotada;
- Avaliar se a representação atual pode ser aperfeiçoada a fim de melhorar o processo;
- Estudar e incorporar técnicas de aprendizado de máquina para o refinamento dos parâmetros e do processo de busca local;
- Estudar variações do MDVRP e outras restrições, como a limitação do tempo total para as rotas.

# Referências

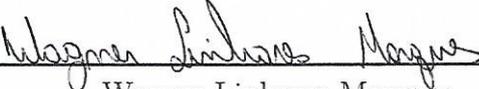
- BÄCK, T. Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms. In: \_\_\_\_\_. New York, NY, USA: Oxford University Press, Inc., 1996. ISBN 0-19-509971-0. Citado na página 19.
- CONTARDO, C.; MARTINELLI, R. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, v. 12, n. 0, p. 129 – 146, 2014. ISSN 1572-5286. Citado na página 17.
- CORDEAU, J.-F.; GENDREAU, M.; LAPORTE, G. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, v. 30, p. 105–119, 09 1997. Citado na página 28.
- CORDEAU, J.-F. et al. Vehicle routing. In: \_\_\_\_\_. [S.l.: s.n.], 2007. v. 14, p. 195–224. Citado 2 vezes nas páginas 13 e 26.
- CREVIER, B.; CORDEAU, J.-F.; LAPORTE, G. The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, v. 176, n. 2, p. 756–773, 2007. Citado na página 13.
- DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. *Management Science*, v. 6, p. 80–91, 10 1959. Citado na página 13.
- GENDREAU, M. et al. Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography. In: GOLDEN, B.; RAGHAVAN, S.; WASIL, E. (Ed.). *The Vehicle Routing Problem: Latest Advances and New Challenges*. [S.l.]: Springer US, 2008, (Operations Research/Computer Science Interfaces, v. 43). p. 143–169. ISBN 978-0-387-77777-1. Citado na página 17.
- KARAKATIČ, S.; PODGORELEC, V. A survey of genetic algorithms for solving multi depot vehicle routing problem. *Applied Soft Computing*, v. 27, n. 0, p. 519–532, 2015. ISSN 1568-4946. Citado na página 17.
- KUO, Y.; WANG, C.-C. A variable neighborhood search for the multi-depot vehicle routing problem with loading cost. *Expert Systems with Applications*, v. 39, n. 8, p. 6949 – 6954, 2012. ISSN 0957-4174. Citado na página 17.
- LUKE, S. *Essentials of Metaheuristics*. second. [S.l.]: Lulu, 2013. Available for free at <http://cs.gmu.edu/~sean/book/metaheuristics/>. Citado na página 19.
- MIRABI, M.; GHOMI, S. F.; JOLAI, F. Efficient stochastic hybrid heuristics for the multi-depot vehicle routing problem. *Robotics and Computer-Integrated Manufacturing*, v. 26, n. 6, p. 564 – 569, 2010. ISSN 0736-5845. 19th International Conference on Flexible Automation and Intelligent Manufacturing. Citado na página 13.
- MONTOYA-TORRES, J. R. et al. A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, v. 79, p. 115 – 129, 2015. ISSN 0360-8352. Citado 2 vezes nas páginas 16 e 17.

- NEDJAH, N.; MOURELLE, L. d. M.; ALBA, E. E. *Parallel Evolutionary Computations*. [S.l.]: Springer, 2006. Citado na página 14.
- OLIVEIRA, F. B. de et al. A cooperative coevolutionary algorithm for the multi-depot vehicle routing problem. *Expert Systems with Applications*, v. 43, p. 117 – 130, 2016. ISSN 0957-4174. Citado 5 vezes nas páginas 13, 16, 17, 18 e 26.
- PRINS, C. Prins, c.: A simple and effective evolutionary algorithm for the vehicle routing problem. *computer & operations research* 31(12), 1985-2002. *Computers & Operations Research*, v. 31, p. 1985–2002, 10 2004. Citado 2 vezes nas páginas 22 e 31.
- RAHIMI-VAHED, A. et al. Fleet-sizing for multi-depot and periodic vehicle routing problems using a modular heuristic algorithm. *Computers & Operations Research*, v. 53, n. 0, p. 9 – 23, 2015. ISSN 0305-0548. Citado na página 17.
- RAY, S. et al. The multi-depot split-delivery vehicle routing problem: Model and solution algorithm. *Knowledge-Based Systems*, v. 71, n. 0, p. 238 – 265, 2014. ISSN 0950-7051. Citado na página 17.
- SALHI, S.; IMRAN, A.; WASSAN, N. A. The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. *Computers & Operations Research*, v. 52, Part B, n. 0, p. 315 – 325, 2014. ISSN 0305-0548. Recent advances in Variable neighborhood search. Citado na página 17.
- VIDAL, T. et al. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, v. 60, n. 3, p. 611–624, 2012. Citado 5 vezes nas páginas 16, 17, 22, 26 e 28.

## TERMO DE RESPONSABILIDADE

Eu, **Wagner Linhares Marques** declaro que o texto do trabalho de conclusão de curso intitulado “*Estudo e Definição de Modelos de Computação Evolucionária para o Problema de Roteamento de Veículos com Múltiplos Depósitos*” é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

João Monlevade, 11 de julho de 2019

  
\_\_\_\_\_  
Wagner Linhares Marques

# DECLARAÇÃO DE CONFORMIDADE

Certifico que o(a) aluno(a) **Wagner Linhares Marques**, autor do trabalho de conclusão de curso intitulado “*Estudo e Definição de Modelos de Computação Evolucionária para o Problema de Roteamento de Veículos com Múltiplos Depósitos*” efetuou as correções sugeridas pela banca examinadora e que estou de acordo com a versão final do trabalho.

João Monlevade, 02 de setembro de 2019.



Fernando Bernardes de Oliveira