

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Algoritmos Genéticos Aplicados ao Problema de Roteamento de
Veículos com Múltiplos Depósitos

Thiago Moreira de Freitas

Itabira, 30 de Agosto de 2023.

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Thiago Moreira de Freitas

Algoritmos Genéticos Aplicados ao Problema de Roteamento de
Veículos com Múltiplos Depósitos

Dissertação submetida ao Programa de Pós-Graduação em
Ciência e Tecnologia da Computação como parte dos requisitos
para obtenção do Título de Mestre em Ciência e Tecnologia
da Computação

Área de Concentração: Matemática da Computação

Orientador: Prof. Dr. Rafael Francisco dos Santos

Agosto de 2023

Itabira - MG

Agradecimentos

Em primeiro lugar agradeço a Deus pela força e coragem para superar as dificuldades nessa longa caminhada.

Aos meus pais, José Alves e Rita, pelo amor incondicional, incentivos em momentos de desânimo e cansaço. O apoio fornecido foi essencial na minha trajetória até o momento.

Ao meu orientador, Rafael Francisco dos Santos, pelo incentivo, conselhos, amizade, paciência e conhecimentos transmitidos.

Aos professores, que tive o prazer de conviver durante o processo do mestrado, pelo apoio e os ensinamentos.

Tente uma, duas, três vezes e se possível tente a quarta, a quinta e quantas vezes for necessário. Só não desista nas primeiras tentativas, a persistência é amiga da conquista. Se você quer chegar a onde a maioria não chega, faça o que a maioria não faz.

Bill Gates

Resumo

O Problema de Roteamento de Veículos ou *Vehicle Routing Problem* (VRP) tem amplas aplicações em logística e transporte com grande importância econômica. VRP é uma generalização de um grande número de problemas de roteamento, que consistem em encontrar o número ideal de rotas, que saem de um único depósito, para atender um conjunto de clientes, minimizando os custos de roteamento e atendendo um conjunto de restrições. O problema de roteamento de veículos de múltiplos depósitos ou *Multi Depot Vehicle Routing Problem* (MDVRP) é uma extensão do VRP, em que há mais de um depósito distribuídos em uma determinada área geográfica. O restante do problema é idêntico ao VRP. Existem vários métodos para a resolução do MDVRP como técnicas exatas, algoritmos aproximativos e heurísticos. Os Algoritmos Genéticos ou *Genetic Algorithms* (GAs) são meta-heurísticas amplamente utilizadas para encontrar soluções para o problema MDVRP devido às características estocásticas dos GAs e a eficiência na resolução de problemas combinatórios e, por esse motivo foi selecionado para ser aplicado nesse trabalho. O algoritmo desenvolvido foi testado utilizando instâncias presentes na literatura e comparado com metodologias existentes, no qual o algoritmo genético encontrou bons resultados e o trabalho trouxe a contribuição na técnica de seleção de clientes que podem realizar trocas entre depósitos. Os resultados alcançados mostram que esse algoritmo pode ser avaliado em empreendimentos reais, possibilitando melhorar a operação de empreendimentos que enfrentam esse tipo de problema, reduzindo custos com transporte, distância, tempo de entrega, serviços, entre outros benefícios.

Abstract

The Vehicle Routing Problem (VRP) has wide applications in logistics and transportation with great economic importance. VRP is a generalization of a large number of routing problems, which consist of finding the optimal number of routes, leaving a single depot, to serve a set of customers, minimizing routing costs and meeting a set of constraints. The Multi Depot Vehicle Routing Problem (MDVRP) is an extension of VRP, in which there is more than one depot distributed in a given geographic area. The rest of the problem is identical to VRP. There are several methods for solving MDVRP such as exact techniques, approximate algorithms and heuristics. Genetic Algorithms (GAs) are meta-heuristics widely used to find solutions to the MDVRP problem due to the stochastic characteristics of GAs and the efficiency in solving combinatorial problems and, for this reason, they were selected to be applied in this work. The developed algorithm was tested using instances present in the literature and compared with existing methodologies, in which the genetic algorithm found good results and the work contributed to the technique of selecting customers who can exchange between deposits. The results achieved show that this algorithm can be evaluated in real projects, making it possible to improve the operation of projects that face this type of problem, reducing transportation costs, distance, delivery time, services, among other benefits.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

p. 11

1 Introdução

p. 13

1.1 Objetivos p. 14

1.2 Resultados e Contribuições p. 15

1.3 Estrutura da Dissertação p. 15

2 Fundamentação Teórica

p. 16

2.1 Problema de Roteamento de Veículos p. 16

2.2 Categorias de MDVRP p. 20

2.3 Trabalhos Relacionados p. 22

2.4 Algoritmos Genéticos p. 24

2.4.1 Codificação dos Indivíduos p. 27

2.4.2 População Inicial p. 27

2.4.3 Função de Aptidão p. 28

2.4.4 Seleção p. 28

2.4.5 Cruzamento (*Crossover*) p. 30

2.4.6 Mutação p. 31

2.5 Algoritmos Genéticos aplicados ao MDVRP p. 32

2.5.1	Representação do Problema	p. 32
2.5.2	População Inicial	p. 32
2.5.3	Seleção	p. 33
2.5.4	Cruzamento/Crossover	p. 33
2.5.5	Mutação	p. 34
2.6	Considerações	p. 35
3	Material e Métodos	p. 36
3.1	Representação do Problema usando Algoritmos Genéticos	p. 36
3.2	Clusterização	p. 39
3.3	População Inicial	p. 42
3.4	Função Objetivo	p. 44
3.5	Elitismo	p. 46
3.6	Seleção	p. 47
3.7	Cruzamento	p. 47
3.8	Mutação	p. 48
3.9	Ejeção de Cliente	p. 51
3.10	2-opt	p. 52
3.11	3-opt	p. 53
3.12	Combinações de Rotas	p. 54
3.13	Fluxograma Algoritmo Genético Swap	p. 55
4	Experimentos e Resultados	p. 57
4.1	Descrição da Base de Dados	p. 57
4.2	Experimentos	p. 57
4.2.1	Experimento 1	p. 59
4.2.2	Experimento 2	p. 63

4.2.3	Experimento 3	p. 64
4.2.4	Experimento 4	p. 66
5	Conclusão	p. 70
5.1	Trabalhos Futuros	p. 71
	Referências	p. 72
	Apêndice A	p. 76

Lista de Figuras

1	Problema de Roteamento de Veículos.	p. 17
2	Problema de roteamento de veículos com múltiplos depósitos.	p. 18
3	Fluxograma de um Algoritmo Genético.	p. 26
4	Codificação binária de indivíduo.	p. 27
5	Codificação de indivíduo no problema de roteamento de veículos.	p. 27
6	Representação de indivíduos em um torneio, ilustrando a seleção de um subconjunto da população de indivíduos e selecionando o indivíduo com melhor aptidão.	p. 29
7	Operação de cruzamento simples com ponto de corte selecionado aleatoriamente	p. 30
8	Operação de cruzamento multiponto com pontos de corte selecionados aleatoriamente	p. 31
9	Operação de mutação simples.	p. 32
10	Representação Lista.	p. 33
11	Cruzamento PMX.	p. 34
12	Cruzamento OX.	p. 35
13	Representação do Problema.	p. 38
14	Clusterização entre clientes e depósitos, baseado em suas posições geográficas.	p. 39
15	Verificação de distância de clientes distantes.	p. 42
16	Método de geração de rotas da população inicial.	p. 44
17	Elitismo no Algoritmo Genético.	p. 46
18	Seleção Torneio.	p. 47

19	Cruzamento.	p. 49
20	Cruzamento Múltiplos Depósitos.	p. 49
21	Mutação de inversão.	p. 50
22	String memória.	p. 51
23	Ejeção de Cliente.	p. 52
24	Operação 2-opt.	p. 53
25	Operação 3-opt.	p. 54
26	Combinação de rotas.	p. 55
27	Fluxograma do algoritmo genético desenvolvido.	p. 56
28	Algoritmo genético sem mutação de clientes entre depósitos comparado com Algoritmo genético com mutação de clientes entre depósitos	p. 64
29	Resultados gerados pelas estruturas para a instância p04	p. 68
30	Resultados gerados pelas estruturas para a instância p12	p. 68

Lista de Tabelas

1	Representação de indivíduos, sua aptidão e porcentagem ocupada em problema da roleta.	p. 29
2	Descrição das instâncias da base de dados.	p. 58
3	Parâmetros utilizados no algoritmo genético desenvolvido	p. 59
4	Cientes diferentes de CD_S20 e CD_S40 comparados com CD_D.	p. 60
5	AG sem mutação de trocas de clientes entre depósitos com distância de clusterização ajustada para 20%.	p. 61
6	Comparação dos resultados sem mutação entre depósitos das estruturas CD_20, CD_S40 e CD_D.	p. 61
7	Média e desvio padrão do GA sem mutação de trocas de clientes entre depósitos.	p. 62
8	Comparação das médias e desvio padrão sem mutação entre depósitos das estruturas CD_20, CD_S40 e CD_D.	p. 62
9	Comparação de GA executado com e sem mutação de clientes entre depósitos	p. 63
10	Melhores resultados do GA com mutação entre depósitos com distância de troca permitida ajustada em 20% e 50%.	p. 65
11	Média e desvio padrão do GA com mutação entre depósitos com distância de troca permitida ajustada em 20% e 50%.	p. 66
12	Melhores resultados do GA com mutação de trocas de clientes entre depósitos.	p. 67
13	Média e desvio padrão do GA com mutação de trocas de clientes entre depósitos.	p. 67

Lista de Abreviaturas e Siglas

ACO	<i>Ant Colony Optimization - Otimização de colônia de formigas</i>
ATN	<i>Automated Transit Networks - Redes de Trânsito Automatizadas</i>
ATN-MDVRP	<i>Automated Transit Networks - Multi Depot Vehicle Routing Problem - Redes de Trânsito Automatizadas com Múltiplos Depósitos</i>
CMDVRPTWA	<i>Problema de Roteamento Colaborativo de Veículos com Vários Depósitos com Atribuição de Janela de Tempo</i>
COMVRP	<i>Problema de Roteamento de Veículos Mistos Fechado-Aberto</i>
COMDMVRP	<i>Problema de Roteamento de Veículos Mistos de Múltiplos Depósitos Fechados e Abertos</i>
COVID-19	<i>Corona virus disease - Doença do Corona virus</i>
FT-SMDVRPTW	<i>Problema de roteamento seletivo de veículos multi-depósito de carga completa sob restrições de janelas de tempo</i>
GA	<i>Genetic Algorithm - Algoritmo Genético</i>
ICO	<i>Imperialist Competitive Algorithm - Algoritmo Competitivo Imperialista</i>
PSO	<i>Particle Swarm Optimization - Otimização de Enxame de Partículas</i>
MDVRP	<i>Multi Depot Vehicle Routing Problem - Problema de Roteamento de Veículos com Múltiplos Depósitos</i>

MD-VRP-HF	<i>Problema de Roteamento de Veículos com Múltiplos Depósitos com Frota Heterogênea</i>
MDVRPLC	<i>Problema Toteamento de Veículos com Múltiplos Depósitos e de Localização de Instalação</i>
MDVRPSPD	<i>Problema de Toteamento de Veículos com Coleta e Entrega Simultâneas</i>
MDVRPSDPIR	<i>Problema de Roteamento de Veículos de Múltiplos Depósitos com Entrega e Coleta Simultâneas e Restrições de Estoque</i>
PDP	<i>Problema de Coleta e Entrega</i>
PSO	<i>Otimização de enxame de partículas</i>
TSP	<i>Traveling Salesman Problem - Problema do Caixeiro Viajante</i>
VRP	<i>Vehicle Routing Problem - Problema de Roteamento de Veículos</i>
SPDP	<i>Problema de Coleta e Entrega Simultânea</i>

1 Introdução

O Problema de Roteamento de Veículos ou *Vehicle Routing Problem* (VRP) é um problema clássico derivado do Problema do Caixeiro Viajante ou *Traveling Salesman Problem* (TSP). O TSP busca encontrar uma rota, que passe por todas as cidades, sem repetir nenhuma delas e retorne para o ponto inicial, de maneira que o caminho seja o mais curto possível. O VRP estende esse problema, na medida em que há um conjunto de clientes ou paradas que devem ser atendidas por uma frota de veículos, que partem de um depósito e ao finalizar a rota retorna para o mesmo. Basicamente é um reflexo de problemas reais, como o de passageiros e entrega de diferentes tipos de mercadorias.

No contexto da pandemia do Covid-19, o estudo se tornou ainda mais importante, pois o transporte de pessoas e bens se tornou um fator crítico para o controle da disseminação do vírus. Uma das principais preocupações em relação ao transporte durante a pandemia era a necessidade de garantir o distanciamento social e reduzir a exposição das pessoas ao vírus. Nesse contexto, houve um aumento significativo na compra de mercadorias de modo online ou via telefone, que sobrecarregou tanto os setores de entrega das empresas, quanto as empresas especializadas em entregas. Com o aumento expressivo da demanda, a otimização das rotas permite maior agilidade nas entregas, redução de custos e a possibilidade de atender a demanda, sem a necessidade de aumentar a frota e o número de funcionários de maneira expressiva. É preciso otimizar as rotas dos veículos de modo a minimizar o tempo de viagem e o número de paradas, reduzindo assim o risco de contágio. O estudo do VRP e seus derivados, também pode contribuir para a redução do impacto econômico da pandemia. Com a necessidade de restrições de circulação, muitas empresas tiveram que adotar o sistema de *delivery* para manter suas atividades, e a otimização das rotas pode reduzir os custos de transporte e aumentar a eficiência das entregas.

A pesquisa desta dissertação, se concentra em uma variação do VRP, o Problema de Roteamento de Veículos com Múltiplos Depósitos, do inglês, *Multi Depot Vehicle Routing Problem* (MDVRP). A principal diferença deste problema é a presença de vários depósitos no qual os veículos iniciam a rota, enquanto o VRP possui apenas um depósito

(KARAKATIČ; PODGORELEC, 2015).

Assim como o VRP, o MDVRP é um problema NP-Difícil, o que torna complexo o processo de encontrar uma solução ótima em um tempo computacional factível (NARASIMHA; KIVELEVITCH; KUMAR, 2012). Embora métodos exatos consigam resolver pequenos problemas com eficiência, em problemas reais que envolvam um número maior de variáveis, os métodos exatos podem necessitar de um tempo computacional proibitivo. O elevado tempo computacional exigido pelos métodos exatos torna a aplicação de algoritmos meta-heurísticos atrativos, pelo fato de encontrarem soluções aceitáveis em menor tempo (KARAKATIČ; PODGORELEC, 2015).

Os algoritmos genéticos são meta-heurísticas inspiradas no processo natural de evolução descrita na teoria da evolução de Charles Darwin. Resultados de pesquisas realizadas na área demonstram que a principal vantagem dos algoritmos genéticos em comparação com outros algoritmos meta-heurísticos, são o seu desempenho e o resultado final obtido levando em consideração as restrições de tempo e limite da capacidade computacional (AROSTEGUI; KADIPASAOGLU; KHUMAWALA, 2006). Portanto, o objetivo da pesquisa nesta dissertação é investigar o uso dos algoritmos genéticos no problema de roteamento de veículos com múltiplos depósitos e propor novos operadores genéticos com o objetivo de trazer respostas melhores ou competitivas com o estado da arte.

1.1 Objetivos

Este trabalho teve como objetivo geral utilizar um Algoritmo Genético para obter resultados significativos que possam contribuir na resolução de problemas de roteamento de veículos com múltiplos depósitos. Para se concluir o objetivo geral, foram realizados os seguintes objetivos específicos:

- A proposição de um modelo de clusterização com o objetivo de melhorar a qualidade da população inicial do Algoritmo Genético;
- Desenvolver novos operadores genéticos, visando melhorar o resultado obtido pela meta-heurística.
- Validar o método proposto por meio de experimentos computacionais utilizando instâncias da literatura.

1.2 Resultados e Contribuições

Ao contrário de estudos que se baseiam em uma metodologia de programação inteira, que demanda um maior poder computacional e tempo para encontrar soluções para o problema de roteamento de veículos com múltiplos depósitos, a metodologia presente neste trabalho implementa um algoritmo genético capaz de encontrar bons resultados em instâncias disponíveis pela literatura e que pode ser avaliado em empreendimentos reais que tem como característica esse modelo de problema. Este trabalho também apresenta duas contribuições para o estado da arte, sendo a primeira uma mudança na criação de listas de trocas de clientes entre depósitos utilizadas em (MAHMUD; HAQUE, 2019; OMBUKI-BERMAN; HANSHAR, 2009), no qual estas listas são utilizadas no operador de mutação entre depósitos. A segunda contribuição é um algoritmo de clusterização denominado Clusterização Swap que expande a clusterização por distância, utilizando as listas de trocas, para alocar os clientes aos depósitos que atenderão às suas demandas.

1.3 Estrutura da Dissertação

O restante deste trabalho está estruturado da seguinte forma. O Capítulo 2 apresenta a fundamentação teórica apresentando as principais características que compõem o problema de roteamento de veículos, categorias de problemas de roteamento de veículos com múltiplos depósitos e o algoritmo genético. Em seguida, o Capítulo 3 descreve a metodologia utilizada para alcançar o objetivo desse trabalho. No Capítulo 4 são descritos os experimentos e resultados alcançados. Finalmente, no Capítulo 5 é apresentada a conclusão e os trabalhos futuros.

2 Fundamentação Teórica

Este capítulo descreve a fundamentação teórica do trabalho e é organizado conforme descrito a seguir. A Seção 2.1 descreve Problema de Roteamento de Veículos ou *Vehicle Routing Problem* (VRP), ressaltando suas principais características. Em seguida, a Seção 2.2 descreve algumas categorias de Problema de Roteamento de Veículos com Múltiplos Depósitos, do inglês, *Multi Depot Vehicle Routing Problem* (MDVRP) existentes na literatura. Logo após, na Seção 2.3 são apresentados os trabalhos relacionados ao estudo do problema de roteamento de veículos com múltiplos depósitos, esses trabalhos discutem e descrevem abordagens utilizadas para encontrar soluções para o problema. A Seção 2.4 descreve o funcionamento dos Algoritmos Genéticos ou *Genetic Algorithms* (GAs). Finalmente, na Seção 2.6 apresenta as considerações sobre o capítulo.

2.1 Problema de Roteamento de Veículos

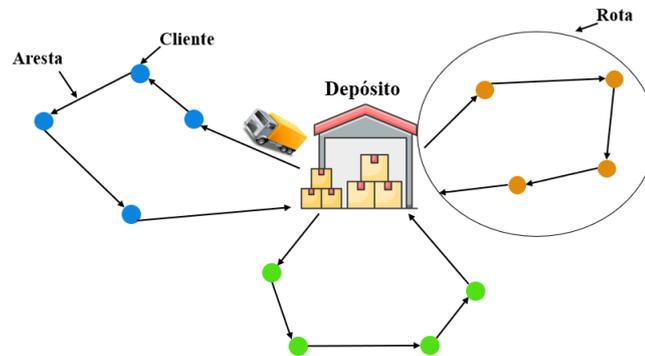
O problema de roteamento de veículos é um problema que representa situações vivenciadas na área de transporte e distribuição, como transportes de pessoas, itens, serviços de transporte, coleta de lixo, entre outros (MOHAMMED et al., 2017). Portanto, uma seleção adequada das rotas a serem percorridas pelos veículos, tem um papel importante para melhorar o planejamento lógico e os interesses econômicos (MOHAMMED et al., 2017).

O problema de roteamento de veículos é derivado do problema clássico do caixeiro viajante (KARAKATIČ; PODGORELEC, 2015). Em seu livro DAVENDRA (2010) descreve o problema do caixeiro viajante ou *Traveling Salesman Problem* (TSP) como, dado um conjunto de cidades e o custo da viagem (ou distância) entre cada um dos pares de cidades possíveis, o problema do caixeiro viajante, busca encontrar o melhor modo possível de visitar todas as cidades e retornar ao ponto de partida minimizando o custo da viagem (ou distância da viagem). Por outro lado, o problema de roteamento de veículos, expande o TSP adicionando um depósito, onde o veículo inicia sua viagem e visita um grupo de cidades (ou clientes, universidades, armazéns, entre outros) e retorna para o depósito que

iniciou a viagem com o menor custo (KARAKATIČ; PODGORELEC, 2015; MOHAMMED et al., 2017; WANG et al., 2016).

LAPORTE (1992) descreve que o VRP é representado por um grafo direcionado $G(V, E)$, onde $V = \{0, 1, \dots, n\}$ representa um conjunto de nós, e E representa o conjunto de arestas. O depósito é denotado com o nó $j = 0$ e os clientes são os nós $j = 1, 2, \dots, n$, onde cada cliente possui uma demanda $d_j > 0$. Cada aresta, representa uma rota do nó i até o nó j . O peso de cada aresta $c_{ij} > 0$ corresponde ao custo (tempo ou distância) de ir do nó i ao no j . Se $c_{ij} = c_{ji}$, o VRP é simétrico, caso contrário, o problema é assimétrico. A Figura 1 ilustra a representação do VRP onde são destacados os clientes, as arestas, o depósito e as rotas.

Figura 1: Problema de Roteamento de Veículos.



Fonte: Elaborada pelo autor.

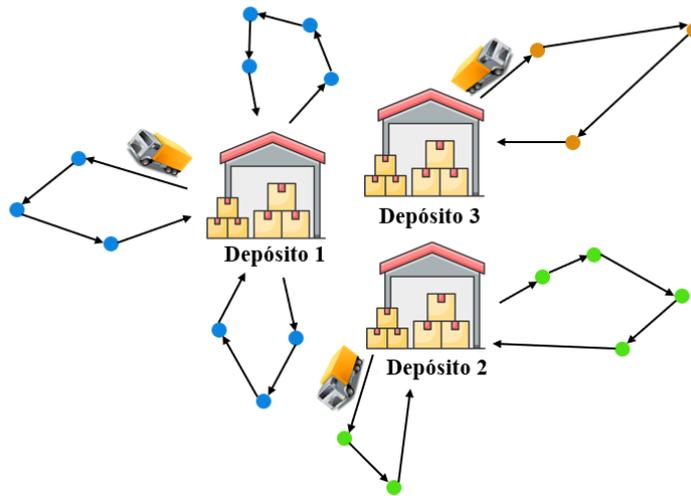
O VRP, tem o objetivo de encontrar um número n de rotas que minimize o custo de viagem, obedecendo as seguintes restrições:

- Cada cliente deve pertencer somente a uma rota;
- Cada rota deve iniciar e terminar no depósito;
- A soma das demandas dos clientes em qualquer rota não pode ser superior à capacidade dos veículos;
- A rota do veículo é calculada adicionando a distância do depósito até o primeiro cliente da rota. Em seguida, é adicionado as distâncias entre os clientes, primeiro com o segundo, segundo com o terceiro, até o último cliente. Por fim, é adicionado a distância do último cliente ao depósito.

O problema de roteamento de veículos com múltiplos depósitos, é uma variação do VRP onde são adicionados vários depósitos (KARAKATIČ; PODGORELEC, 2015) a Figura

2 apresenta uma ilustração desse tipo de problema. MDVRP, assim como o VRP, é um problema NP-difícil de otimização combinatória portanto é difícil encontrar sua solução ótima em um tempo computacional factível (KARAKATIĆ; PODGORELEC, 2015; NARASIMHA; KIVELEVITCH; KUMAR, 2012).

Figura 2: Problema de roteamento de veículos com múltiplos depósitos.



Fonte: Elaborada pelo autor.

De acordo com RENAUD; LAPORTE; BOCTOR (1996), o MDVRP pode ser descrito como um grafo $G = (V, E)$, onde V é o conjunto de nós e E é o conjunto de arestas conectando cada par de nós. O conjunto V é subdividido em dois subconjuntos: $V_c = v_1, v_2, \dots, v_N$ que é o conjunto de clientes a serem atendidos; e $V_d = v_{N+1}, v_{N+2}, \dots, v_M$ que é o conjunto de depósitos. Cada cliente $v_i \in V_c$ tem uma demanda não negativa d_i . Cada aresta pertencente ao conjunto E tem associado um custo, distância ou tempo de viagem c_{ij} . Além de G , há um total de K veículos, cada um com capacidade p_k . T representa o custo máximo da rota.

De acordo com KULKARNI; BHAVE (1985), um modelo matemático do MDVRP requer que a definição das variáveis de decisão binárias x_{ijk} sejam iguais a 1 se o par de nós i e j estiverem na rota do veículo k , e 0 caso contrário. As variáveis auxiliares y_i são necessárias para eliminação de sub-rota. O problema é modelado conforme as equações 2.1 a 2.11.

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

Sujeito a:

$$\sum_{i=1}^{N+M} \sum_{k=1}^K x_{ijk} = 1 \quad (j = 1, \dots, N) \quad (2.2)$$

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

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

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

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

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

$$\sum_{j=N+1}^{N+M} \sum_{i=1}^N x_{ijk} \leq 1 \quad (k = 1, \dots, K) \quad (2.8)$$

$$y_i - y_j + (M + N)x_{ijk} \leq N + M - 1$$

Para $1 \leq i \neq j \leq N$ e $1 \leq k \leq K$ (2.9)

$$x_{ijk} \in \{0, 1\} \quad \forall i, j, k \quad (2.10)$$

$$y_i \in \{0, 1\} \quad \forall i \quad (2.11)$$

Nesta formulação, a função objetivo é minimizar o custo total, as restrições (2.2) e (2.3) garantem que cada cliente seja atendido por apenas um veículo. A continuidade da rota é representada pela restrição (2.4). Os conjuntos de restrições (2.5) e (2.6) são a capacidade do veículo e as restrições de custo total da rota. A disponibilidade do veículo é verificada pelas restrições (2.7) e (2.8) e a eliminação de sub-rota é fornecida pelas restrição (2.9). Nesta formulação, assume-se que a demanda total em cada nó é menor ou no máximo igual à capacidade de cada veículo.

2.2 Categorias de MDVRP

A seguir são apresentadas algumas categorias de problemas de roteamento de veículos com múltiplos depósitos encontrados na literatura, ilustrando suas principais características dentro deste segmento de problema:

- **Problema de roteamento de veículos com coleta e entrega (MDVRPPD):** É uma extensão do problema de roteamento de veículo com múltiplos depósitos, que surge em aplicações do mundo real, onde envolve a determinação de rotas para veículos que precisam coletar produtos em um conjunto de locais de coleta e entregar em um conjunto de locais de entrega. As operações de coleta e entrega são levadas em consideração para satisfazer a restrição de capacidade do veículo com o objetivo de minimizar a distância total percorrida (KOULAEIAN et al., 2015);
- **Problema de roteamento de veículos com múltiplos depósitos com janela de tempo (MDVRPTW):** Essa categoria de problema projeta rotas de menor custo de cada depósito para um conjunto de clientes distribuídos geograficamente, enquanto satisfaz as restrições da capacidade dos veículos e da janela de tempo dos clientes, onde as entregas ou coletas devem ser realizadas dentro de um determinado intervalo de tempo (RABBANI et al., 2018);
- **Problema de roteamento seletivo de veículos multi-depósito de carga completa sob restrições de janelas de tempo (FT-SMDVRPTW):** Trata-se de um problema de roteamento seletivo de veículos com carga total de caminhões com janelas de tempo, frota heterogênea e múltiplos depósitos. O problema consiste em selecionar as encomendas e determinar a sua ordem de passagem para a entrega, coleta e os veículos a utilizar. O problema pressupõe que um ponto de coleta corresponda apenas a um ponto de entrega e um pedido seja atendido apenas pelo mesmo veículo. Há um lucro associado a cada visita de pedido. O objetivo é maximizar os lucros da empresa de logística (BOUYAHYIOUY; BELLABDAOUI, 2016);
- **Problema de roteamento de veículos com múltiplos depósitos com frota heterogênea (MD-VRP-HF):** O problema é uma generalização do VRP capacitado. Para MD-VRP-HF o objetivo é minimizar os custos totais e tomar decisões sobre a alocação ótima de clientes para depósito, escolha do veículo para visitar cada cliente e construir um sequência ideal de entrega de vários produtos a clientes diferentes (SINGH; L.; PUNDIR, 2019);

- **Problema roteamento de veículos com múltiplos depósitos e de localização de instalação (MDVRPLC):** Neste problema é feita a junção do problema de localização de instalação com o problema de roteamento de veículos com múltiplos depósitos. Na primeira fase, o problema consiste em selecionar em quais locais instalar um depósito e atribuir um e apenas um depósito a cada cliente. Na segunda fase, para cada depósito, é buscado solucionar um problema de roteamento de veículos, vários veículos homogêneos saem de um único depósito e retornam ao mesmo. Cada rota deve respeitar a capacidade de cada veículo. O objetivo deste problema é minimizar as distâncias totais aos percursos realizados (OUDOUAR; FALLAHI; ZAOU, 2019);
- **Problema de roteamento de veículos com coleta e entrega simultâneas (MDVRPSPD):** O problema de coleta e entrega (PDP) é uma das principais classes do problema de roteamento de veículos, no qual um conjunto de rotas é projetado para atender às demandas de coleta e entrega dos clientes. No problema de Coleta e Entrega Simultânea (SPDP), uma frota de veículos proveniente de um centro de distribuição deve entregar algumas mercadorias aos clientes e, ao mesmo tempo, recolher de volta o excesso. Esse problema surge especialmente no contexto da logística reversa, onde as empresas são cada vez mais confrontadas com a tarefa de gerenciar o fluxo reverso de produtos acabados ou matérias-primas. O MDVRPSPD é a junção do problema de roteamento de múltiplos depósitos com o problema de coleta e entrega simultânea (KOULAEIAN et al., 2015);
- **Problema de roteamento colaborativo de veículos com múltiplos depósitos com atribuição de janela de tempo (CMDVRPTWA):** É uma variação do problema de roteamento de veículos com múltiplos depósitos. CMDVRPTWA combina o Problema de Roteamento de Veículos com Múltiplos Depósitos com Janela de Tempo ou *Multi Depot Vehicle Routing Problem with Time Windows* (MDVRPTW) com a atribuição de janela de tempo ou *Time Window Assignment* (TWA) (MDVRPTWA) e problemas de alocação de lucros. O CMDVRPTWA incorpora TWA e integra um mecanismo de colaboração para projetar redes de logística cada vez mais eficientes, minimizando o custo operacional agregado e reduzindo a carga de tráfego. A rede de distribuição colaborativa multi-depósito com TWA é um projeto de rede de distribuição composto por múltiplos centros de distribuição e diversos clientes. A colaboração entre as instalações de logística (centros de distribuição) na rede de distribuição e a TWA para os clientes é projetada para melhorar a eficiência operacional da rede de distribuição e garantir que os produtos sejam entregues aos clientes no prazo (WANG et al., 2020);

- **Problema de roteamento de veículos de múltiplos depósitos com entrega e coleta simultâneas e restrições de estoque (MDVRPSDPIR):** A partir de múltiplos depósitos com níveis de estoque definidos, são encontradas rotas com custos menores, de modo que as demandas conhecidas de entrega e coleta de um conjunto de clientes sejam satisfeitas e as restrições de capacidade, estoque e depósitos, respectivamente, não sejam violadas (BOUANANE; BENADADA; BENCHEIKH, 2018);
- **Problema de roteamento de veículos mistos de múltiplos depósitos fechados e abertos (COMDMVRP):** Este tipo de problema pode ser visto como, a combinação do MDVRP com o problema de roteamento de veículos mistos fechado-aberto (COMVRP). Sendo considerado a frota heterogênea de veículos, que são diferentes em capacidade e comprimento máximo da rota. Nesse caso, ele usa da característica do COMVRP onde alguns veículos não precisam retornar ao depósito, mas o restante dos veículos deve retornar ao depósito de onde partiram (AZADEH; FARROKHI-ASL, 2017).

2.3 Trabalhos Relacionados

Em problemas de otimização o objetivo é obter uma solução ótima, ou uma boa solução. Atualmente, existem três tipos de métodos que podem ser aplicados para encontrar soluções para os problemas, sendo os métodos exatos, heurísticos e de aproximação. Os métodos de soluções exatas, garantem, caso exista, a obtenção de soluções ótimas globais, mas a quantidade total de tempo de computação pode aumentar exponencialmente. Em relação aos algoritmos de aproximação, eles garantem que a solução encontrada não é pior que uma fração pré-determinada do ótimo. Por fim, temos o método de pesquisa baseado em heurística e meta-heurísticas, que podem obter boas soluções, sem garantia de otimalidade, dentro de um tempo razoável e são comumente usadas na prática (WANG et al., 2016). Nesta seção, será mencionado alguns estudos que apresentam abordagens heurísticas para a solução de problemas de roteamento de veículos com múltiplos depósitos.

A abordagem proposta por LI; SHI; FAN (2019) baseada no algoritmo *Imperialist Competitive Algorithm* (ICA), que simula a competição entre impérios para encontrar a melhor solução para um problema. Os resultados mostraram que a abordagem proposta pelos autores é eficaz na resolução do problema de roteamento de veículos com múltiplos depósitos, produzindo soluções de qualidade em um tempo de execução razoável.

Os trabalhos de (MESA; BIEDERMANN, 2006; MOKHTAR; ELHOSENY; HASSANIEN,

2014; RENAUD; LAPORTE; BOCTOR, 1996; LI et al., 2018; SHANG et al., 2015) utilizam a técnica de Busca Tabu, essa heurística utiliza uma lista tabu para evitar retornar a soluções visitadas anteriormente e, assim, explorar novas regiões do espaço de busca. Nos estudos, a técnica é combinada com busca local para encontrar soluções para o MDVRP.

O trabalho de (ZHANG et al., 2019) apresenta uma abordagem de otimização baseada em colônia de formigas (ACO) com objetivo de minimizar a pior rota (min-max VRP). ACO é uma técnica heurística inspirada no comportamento das formigas em busca de alimentos. A técnica consiste em criar uma colônia de formigas artificiais que exploram diferentes soluções para o problema. A técnica foi capaz de encontrar soluções de alta qualidade para as instâncias de teste em um tempo de execução razoável. Além do trabalho citado, existem outros estudos feitos utilizando a metodologia de colônia de formigas (WEN et al., 2011; HE; ZHANG; WANG, 2017; JIANG et al., 2017; WANG; XU; XU, 2016).

Outra heurística utilizada para resolver MDVRP são os algoritmos de enxame de partículas, ou *Particle Swarm Optimization* (PSO) (LIU; LIANG; YANG, 2010; ZHANG; HUANG; ZHANG, 2011; GAO et al., 2016). PSO é uma técnica de otimização baseada em modelos inspirados no comportamento de enxames de animais, como aves ou peixes. No PSO, uma população de partículas é inicializada aleatoriamente no espaço de busca e cada partícula representa uma possível solução do problema. A cada iteração, as partículas são movidas em direção as melhores soluções encontradas até o momento, tanto em termos de soluções individuais quanto globais. A escolha da direção são definidas através de equações matemáticas que combinam a posição atual da partícula com sua melhor posição anterior e a melhor posição global encontrada por qualquer partícula no enxame.

WANG et al. (2016) apresentam uma solução para este problema utilizando técnicas de algoritmos genéticos adaptativos que ajustam a probabilidade de seleção dos indivíduos da população de acordo com o valor de sua função de aptidão. Além disso, o algoritmo incorpora um mecanismo de busca local para melhorar ainda mais a qualidade das soluções. Na literatura existem vários trabalhos que utilizam a abordagem que combina o algoritmo genético com uma busca local (BAI; GAO, 2010; GAO; JIA; LIU, 2016; TAUFIK; RAMADHAN; NABABAN, 2015; ZHANG et al., 2019).

Outra abordagem utilizando algoritmos genéticos para resolver o problema de roteamento de veículos com múltiplos depósitos é utilizando clusterização da população inicial com listas de trocas de clientes entre depósitos. A clusterização de clientes para um depósito é determinada pela distância euclidiana, onde os clientes são alocados ao depósito mais próximo. A partir desse processo de agrupamento é criada uma lista de clientes que

podem realizar trocas entre depósitos, geralmente a lista é composta em clientes limítrofes. A lista criada é utilizada na operação de mutações entre depósitos, onde um cliente é transferido para outro depósito e modifica a estrutura de representação dos indivíduos. Este tipo de mutação é aplicada sempre quando atendem a algum critério como o número de gerações, número de gerações sem melhoras e outros. Essa abordagem de solução para o MDVRP resolve o problema como se cada estrutura de clientes agrupados por depósitos fosse um VRP. Portanto, cada estrutura poderia ser executada em um algoritmo genético e o conjunto de todos os resultados seria a solução do MDVRP (MAHMUD; HAQUE, 2019; OMBUKI-BERMAN; HANSHAR, 2009).

Como apresentado, diversas abordagens heurísticas são utilizadas para resolver problemas de roteamentos de veículos com múltiplos depósitos, como algoritmos genéticos, busca tabu, enxame de partículas, colônia de formigas, entre outras. Entre essas abordagens, o algoritmo genético é um método eficaz para resolver o MDVRP (WU et al., 2018; SONG; WEN; ZHANG, 2018; WANG et al., 2019). Os algoritmos genéticos são técnicas de otimização que tem sido amplamente utilizada para resolver problemas de roteamento de veículos com múltiplos depósitos (WANG et al., 2016). Podendo encontrar soluções boas o suficiente para o problema em um tempo razoável, mesmo quando o número de clientes e veículos é grande e a complexidade do problema é alta. Além disso, os algoritmos genéticos são capazes de explorar várias soluções simultaneamente e é capaz de escapar de mínimos locais. Portanto, os algoritmos genéticos são técnicas eficazes para resolver problemas de roteamento de veículos com múltiplos depósitos e foi a técnica escolhida para o desenvolvimento desta dissertação. Na próxima seção, será apresentado em detalhes o funcionamento do Algoritmo Genético (GA).

2.4 Algoritmos Genéticos

Os algoritmos genéticos são meta-heurísticas inspiradas no processo de evolução natural apresentada pelo inglês Charles Darwin em seu livro, *A Origem das Espécies* (DARWIN, 1859). Como na evolução natural, os algoritmos genéticos criam uma população de indivíduos e utilizam os mecanismos de seleção natural, reprodução, cruzamento e mutação para simular a evolução da população (S; JOSEPH; JAYAKUMAR, 2015; KOULAEIAN et al., 2015). Devido à sua grande flexibilidade, os algoritmos genéticos são uma excelente abordagem para resolver problemas complexos e gerar soluções de alta qualidade (AZADEH; FARROKHI-ASL, 2017; OUDOUAR; FALLAHI; ZAOUI, 2019).

MOHAMMED et al. (2017) citam que os algoritmos genéticos são eficientes para:

- Encontrar uma solução quando o espaço de busca da solução for grande e quando um método exato não conseguir encontrar uma solução em tempo adequado;
- Lidar com problemas de múltiplas restrições;
- Resolver um problema quando há tempo ou recursos limitados para o problema;
- Encontrar uma solução aproximada.

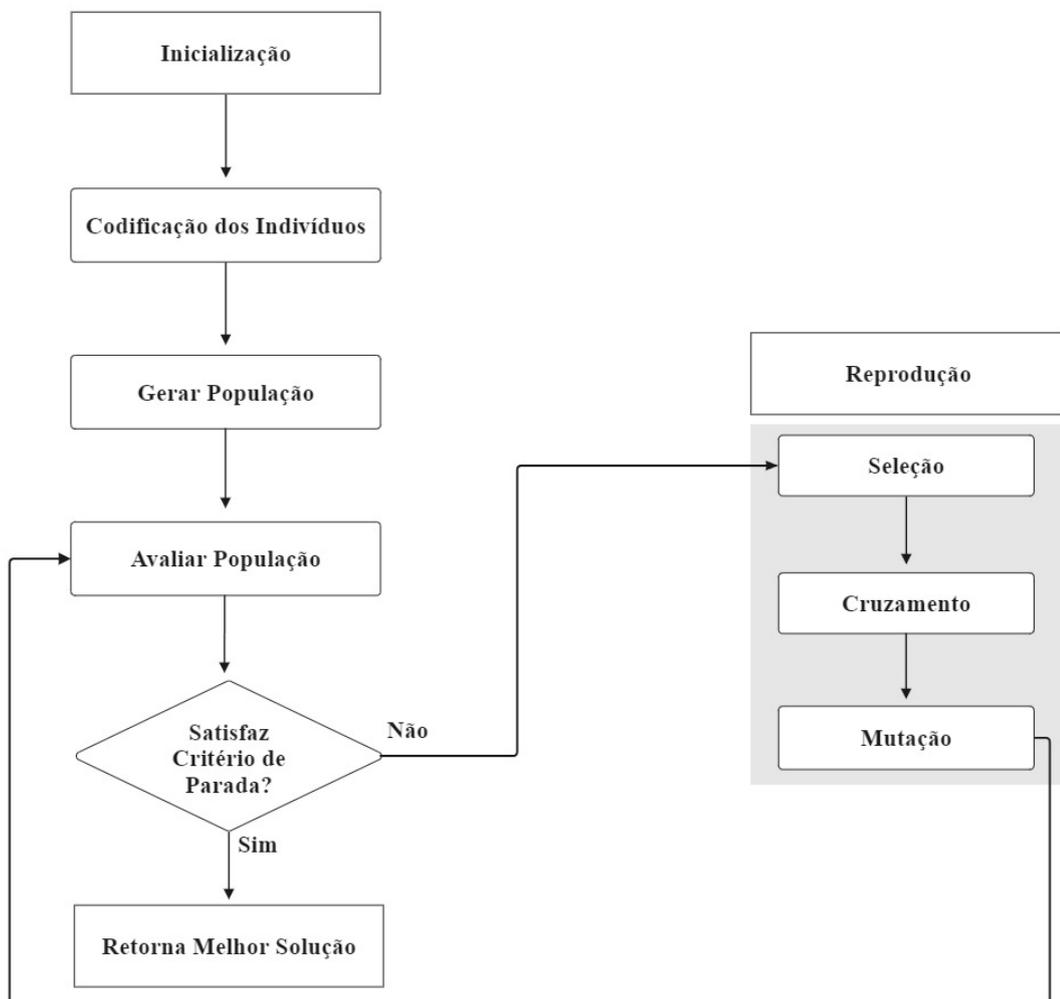
Como os algoritmos genéticos se inspiram nos princípios da seleção natural e evolução biológica, onde indivíduos mais aptos tendem a transferir suas características para os indivíduos da próxima geração. A seguir será destacado algumas características destes processos:

- **Genótipo:** características de um indivíduo.
- **Indivíduo:** um “ser” composto por vários genes que formam seus cromossomos. A representação do indivíduo também é referida como genótipo.
- **População:** conjunto de indivíduos da mesma espécie que vivem em um ambiente.
- **Aptidão:** representa quão adaptado um indivíduo está ao meio que vive, ou seja, possuem características mais adaptadas ao ambiente que vive.
- **Seleção natural:** é o processo pelo qual os organismos mais aptos ao meio, tendem a sobreviver, se reproduzir e ter mais descendentes.
- **Reprodução:** é a combinação dos genes de dois indivíduos, através do cruzamento, para produzir novos indivíduos que irão compor as próximas gerações da população.
- **Mutação:** é a mudança de características genéticas dos indivíduos, onde genes são modificados introduzindo diversidade na população.
- **Evolução:** é a transferência das características hereditárias de uma geração para outra da população.

A Figura 3 indica o fluxo de funcionamento básico de um Algoritmo Genético, baseado nas características anteriores, que pode ser utilizado em diversas aplicações. O algoritmo genético se inicia com a codificação dos indivíduos, geralmente apresentado por um vetor

binário, posteriormente é formando uma população inicial, usualmente feita de forma aleatória. Logo após é utilizada a função de aptidão onde são avaliados os indivíduos da população. Posteriormente é feita a seleção que é o processo de selecionar os indivíduos da população com base em sua aptidão para acasalarem e produzir novos descendentes. O próximo passo da meta-heurística é o cruzamento onde é simulado o acasalamento entre dois indivíduos com o objetivo de produzir filhos. Em seguida é realizada a mutação, e por último deve-se estabelecer um critério de parada, todo processo de evolução deve terminar em algum ponto (MIDAUI; QBADOU; MANSOURI, 2021; AZADEH; FARROKHI-ASL, 2017; OUDOUAR; FALLAHI; ZAOUI, 2019; RABBANI et al., 2018).

Figura 3: Fluxograma de um Algoritmo Genético.



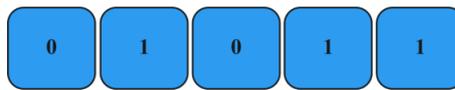
Fonte: Elaborada pelo autor.

Nas próximas seções serão discutidas cada parte dos algoritmos genéticos.

2.4.1 Codificação dos Indivíduos

Inicialmente no algoritmo genético é necessário modelar o problema, onde cada indivíduo representa uma solução do problema, frequentemente sendo representado por um vetor. Um modo de representação é a codificação binária, nela o indivíduo é preenchido com o valor de 0 ou 1, onde 1 indica que a característica da solução está presente e 0 não, ilustrado na Figura 4.

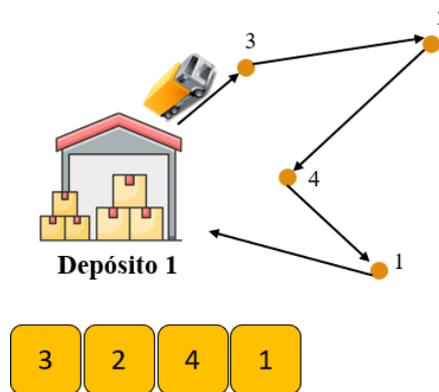
Figura 4: Codificação binária de indivíduo.



Fonte: Elaborada pelo autor.

Outro modo de codificação, utilizado em problemas de roteamentos de veículos, é a utilização de vetores para armazenar a sequência de clientes que compõe uma rota. A Figura 5 ilustra a representação descrita, nesse exemplo é apresentada a rota que atende os clientes [3, 2, 4 e 1].

Figura 5: Codificação de indivíduo no problema de roteamento de veículos.



Fonte: Elaborada pelo autor.

2.4.2 População Inicial

Os Algoritmos Genéticos iniciam a sua execução a partir de uma população inicial, podendo esta ser definida de forma aleatória. No caso dos indivíduos com representação binária, cada cromossomo é definido de forma aleatória assumindo 0 ou 1. A vantagem de utilizar a população inicial, gerada de forma aleatória, é que a população conterà uma boa diversidade que pode representar uma boa região do espaço de busca.

Uma outra técnica para criar a população inicial é utilizando algum algoritmo guloso. Neste caso, os indivíduos são gerados usando uma heurística gulosa que busca soluções viáveis para o problema. Essa estratégia pode ser interessante para problemas onde há alguma informação disponível sobre as soluções viáveis e onde a qualidade da solução inicial é importante para a convergência do algoritmo (LIU; WANG, 2014).

Outra abordagem de geração da população inicial, são as soluções pré-existentes também conhecida como seeds, ou sementes. Em alguns casos, é possível utilizar soluções pré-existentes para gerar a população inicial. Por exemplo, se já houver uma solução conhecida para o problema, é possível utilizá-la como base para gerar outras soluções por meio de perturbações aleatórias ou por meio de outras heurísticas.

A combinação entre as abordagens acima também são utilizadas com o objetivo de proporcionar à população inicial as vantagens de cada técnica.

2.4.3 Função de Aptidão

A função de aptidão é o método de avaliação de cada indivíduo, também conhecida como *fitness*. Esta função define o quão apto uma solução está, baseada em uma função de maximização ou minimização relacionada ao problema que será resolvido. A função de aptidão é aplicada na escolha dos indivíduos no processo de seleção (KARAKATIČ; PODGORELEC, 2015).

2.4.4 Seleção

O processo de seleção é utilizado para escolher quais indivíduos irão passar suas características para a próxima população. Como na teoria de Darwin, os indivíduos com maior aptidão, apresentam maior probabilidade de repassar seus genes através da reprodução (LU; WU; ZHOU, 2017). A etapa representa o processo onde os pais são selecionados baseado em sua aptidão, para realizarem o acasalamento e produzirem uma nova geração (OUDOUAR; FALLAHI; ZAOUI, 2019).

Uma das técnicas utilizadas é a seleção por roleta, onde cada indivíduo da população ocupa uma parte da roleta dimensionada em proporção à sua função de aptidão (GOLDBERG, 1989). Portanto, indivíduos com maiores aptidões ocupam maiores partes da roleta, aumentando a probabilidade de serem selecionados (FITRIANA; MOENGIN; KUSUMANINGRUM, 2019; AZADEH; FARROKHI-ASL, 2017).

A Tabela 1, apresenta uma lista de indivíduos e suas respectivas aptidões, onde quanto maior a aptidão, maior será a porcentagem de ocupação de uma fatia da roleta.

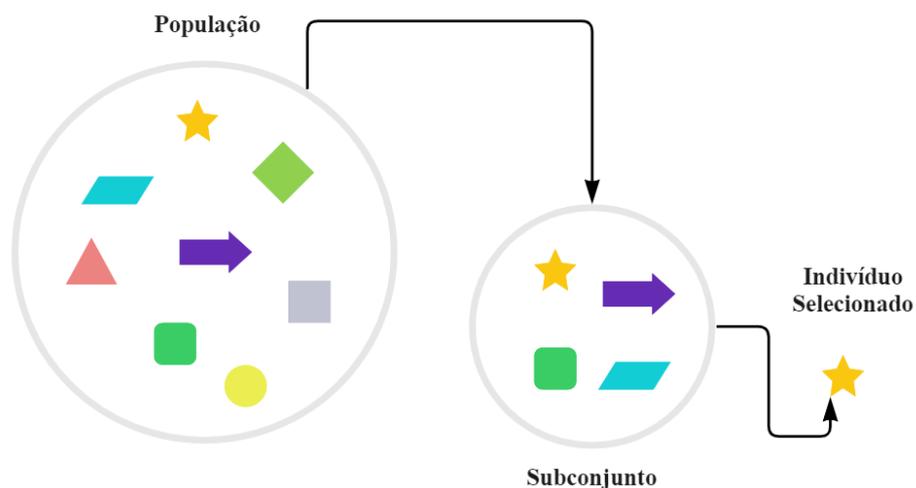
Tabela 1: Representação de indivíduos, sua aptidão e porcentagem ocupada em problema da roleta.

N°	Indivíduo	Aptidão	% do Total
1	01101	169	14,40
2	11000	576	49,20
3	01000	64	5,50
4	10011	361	30,90
Total		1170	100,00

Fonte: Elaborada pelo autor, adaptado de GOLDBERG (1989).

No método de seleção por torneio, vários indivíduos são selecionados aleatoriamente da população e estão envolvidos em um torneio. O indivíduo com melhor aptidão do grupo selecionado vence o torneio (S; JOSEPH; JAYAKUMAR, 2015). A Figura 6, ilustra o funcionamento do método de seleção de torneio. Inicialmente são selecionados aleatoriamente um subconjunto de indivíduos pertencente a população, logo após é realizado o torneio com os participantes deste subconjunto e finalmente de acordo com uma probabilidade de acordo com a aptidão é selecionado o indivíduo que vence o torneio.

Figura 6: Representação de indivíduos em um torneio, ilustrando a seleção de um subconjunto da população de indivíduos e selecionando o indivíduo com melhor aptidão.



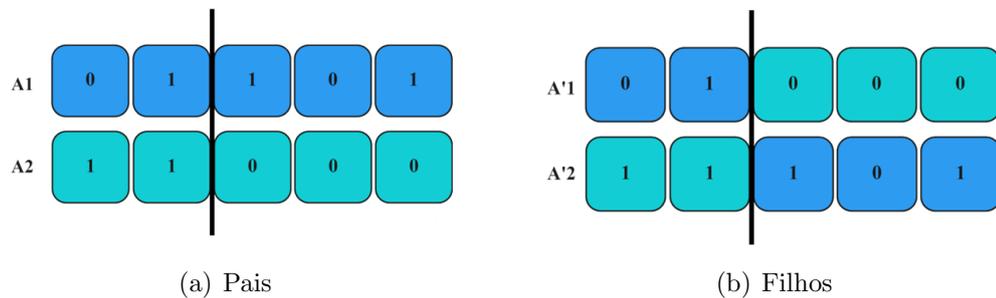
Fonte: Elaborada pelo autor.

2.4.5 Cruzamento (Crossover)

Cruzamento ou *crossover* é o processo que simula o acasalamento entre indivíduos com o objetivo de produzir filhos para a próxima geração. Como na natureza, esses filhos adquirem características de ambos os pais. O objetivo deste procedimento é tentar encontrar filhos mais aptos através da combinação das características dos pais (LU; WU; ZHOU, 2017; OUDOUAR; FALLAHI; ZAOU, 2019; BOUYAHYIOUY; BELLABDAOUI, 2016).

Existem vários tipos de cruzamento, no cruzamento simples, uma posição inteira k é selecionada aleatoriamente entre 1 e o comprimento do vetor l menos um $[1, l - 1]$ dos pais. A partir desta seleção, dois novos indivíduos (filhos) são criados trocando todos os valores entre as posições $k + 1$ e l , inclusive. Por exemplo, considere os indivíduos A1 e A2 da Figura 8(a), onde o valor de K obtido é 2 (conforme indicado pelo símbolo separador |). O cruzamento resultante apresentado na Figura 8(b) produz dois novos indivíduos (filhos), A'1 e A'2, que farão parte da nova geração (GOLDBERG, 1989).

Figura 7: Operação de cruzamento simples com ponto de corte selecionado aleatoriamente



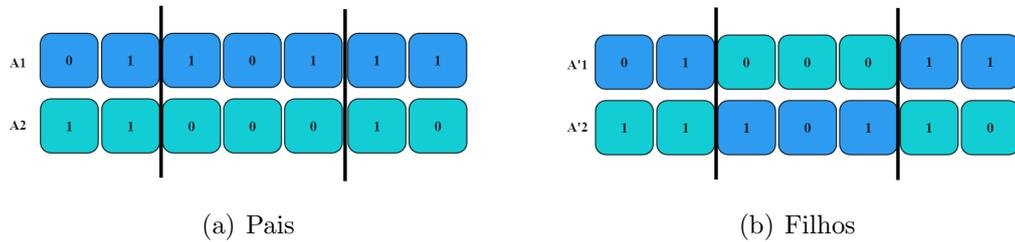
Fonte: Elaborada pelo autor.

O cruzamento multiponto é uma técnica utilizada em algoritmos genéticos que envolve a combinação de informações genéticas de dois indivíduos para gerar um novo indivíduo na população. Essa técnica é uma variação do cruzamento simples e é especialmente útil para problemas em que a ordem das informações genéticas é importante e a diversidade genética é crucial para a evolução da solução.

No cruzamento multiponto, os pontos de cruzamento são selecionados aleatoriamente nos cromossomos dos pais para o cruzamento. Esses pontos de cruzamento definem as seções do cromossomo que serão trocadas entre os indivíduos, gerando novos cromossomos que combinam características de ambos os pais (CHEN et al., 2017).

Na Figura 8, ilustra dois pais com sequências de genes A1 e A2. Primeiro, é selecionado dois pontos de corte aleatórios, que dividem as sequências em três partes diferentes

Figura 8: Operação de cruzamento multiponto com pontos de corte selecionados aleatoriamente



Fonte: Elaborada pelo autor.

(conforme indicado pelo símbolo separador |).

Para cada filho, copia-se uma parte da sequência de um dos pais, alternando as informações genéticas entre os pais, de acordo com os pontos de corte escolhidos, como ilustrado na Figura 8 (b), onde a primeira parte do filho A'1 é herdada do pai A1, a segunda parte do pai A2 e a terceira parte novamente do pai A1.

O resultado final do cruzamento de A1 e A2 são dois novos filhos A'1 e A'2 com informações genéticas combinadas de seus pais, em diferentes segmentos da sequência, aumentando a diversidade genética e ajudando a evitar a convergência prematura.

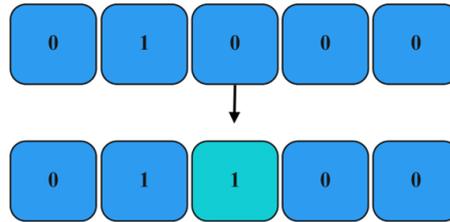
O cruzamento é aplicado no processo de reprodução de uma nova população de acordo com a taxa definida como parâmetro do GA.

2.4.6 Mutação

Mutação é um operador genético usado para preservar a diversidade do GA, ajudando a evitar ficar preso em um ótimo local (AZADEH; FARROKHI-ASL, 2017). O processo de mutação consiste em alterar um ou alguns pontos de um indivíduo com uma probabilidade relativamente pequena, para que o indivíduo não perca todas as características de seus progenitores, consequentemente podendo melhorar a capacidade de pesquisa (LU; WU; ZHOU, 2017). A probabilidade é definida pela taxa de mutação.

No GA simples, a mutação é baseada em uma posição de um vetor. Na codificação binária, isso significa simplesmente alterar o valor do gene de 1 para 0 e vice-versa (GOLDBERG, 1989). Na Figura 9 indica a operação de mutação simples em um vetor binário.

Figura 9: Operação de mutação simples.



Fonte: Elaborada pelo autor.

2.5 Algoritmos Genéticos aplicados ao MDVRP

Vários trabalhos tem aplicado algoritmos genéticos para resolver o problema MDVRP (KARAKATIČ; PODGORELEC, 2015). Nas próximas seções serão discutidas algumas representações utilizadas, algumas formas de criação da população inicial e alguns operadores genéticos utilizados na literatura.

2.5.1 Representação do Problema

Na literatura geralmente os indivíduos são representados por um vetor. Onde o vetor é preenchido com números que identificam as paradas e essas paradas são visitadas uma após a outra (S; JOSEPH; JAYAKUMAR, 2015; OPOKU-AMANKWAAH; OMBUKI-BERMAN, 2017; LU; WU; ZHOU, 2017; MAHMUD; HAQUE, 2019; LI; LI; PARDALOS, 2014; BOUYAHYIOUY; BELLABDAOUI, 2016). Existe uma representação onde o vetor é dividido em duas partes, onde uma parte representa os depósitos e a outra parte representa a sequencia de clientes que os depósitos atendem (WANG et al., 2016). Outro tipo de representação é utilizando matrizes para representar as rotas (BOUANANE; BENADADA; BENCHEIKH, 2018). Uma outra representação é feita por listas, onde existe uma lista para o cliente, depósito e o veículo (SINGH; L.; PUNDIR, 2019). A Figura 10 indica a representação de lista para um problema de MDVRP onde cada posição da lista corresponde ao identificador do cliente, depósito e veículo. Por exemplo, o depósito de identificador 5 atende os clientes [3, 1 e 2] e o veículo com identificador 3 pertence ao depósito 5 e atende os clientes [3, 1 e 2].

2.5.2 População Inicial

Em relação à geração da população inicial, as mais utilizadas são as estratégias que geram os indivíduos aleatoriamente sem quaisquer restrições, que utilizam heurísticas ou a combinação das duas técnicas (BOUANANE; BENADADA; BENCHEIKH, 2018; LU; WU;

Figura 10: Representação Lista.

Cliente	4	5	6	7	3	1	2
Depósito	11	11	11	11	5	5	5
Veículo	1	1	1	1	3	3	3

Fonte: Elaborada pelo autor.

ZHOU, 2017; SINGH; L.; PUNDIR, 2019; HAERANI et al., 2017; LI; LI; PARDALOS, 2014; BOUYAHYIOUY; BELLABDAOUI, 2016). Alguns trabalhos utilizam estratégias de agrupamentos/clusterizações, onde o agrupamento é feito utilizando a distância do cliente em relação ao depósito (S; JOSEPH; JAYAKUMAR, 2015) ou aplicando outras abordagens como o algoritmo K-Means (MAHMUD; HAQUE, 2019; WANG et al., 2020) e o algoritmo K-NN (BOUANANE; BENADADA; BENCHEIKH, 2018).

2.5.3 Seleção

Os métodos de seleção mais utilizados encontrados na literatura foram os métodos do tipo torneio (S; JOSEPH; JAYAKUMAR, 2015; BOUANANE; BENADADA; BENCHEIKH, 2018; OPOKU-AMANKWAAH; OMBUKI-BERMAN, 2017; MAHMUD; HAQUE, 2019; LI; LI; PARDALOS, 2014) e roleta (LU; WU; ZHOU, 2017; SINGH; L.; PUNDIR, 2019; HAERANI et al., 2017; BOUYAHYIOUY; BELLABDAOUI, 2016).

Outra abordagem de seleção é o ranqueamento (WANG et al., 2016, 2020), onde os indivíduos da população são ordenados de acordo com sua aptidão e selecionados de acordo com o ranking.

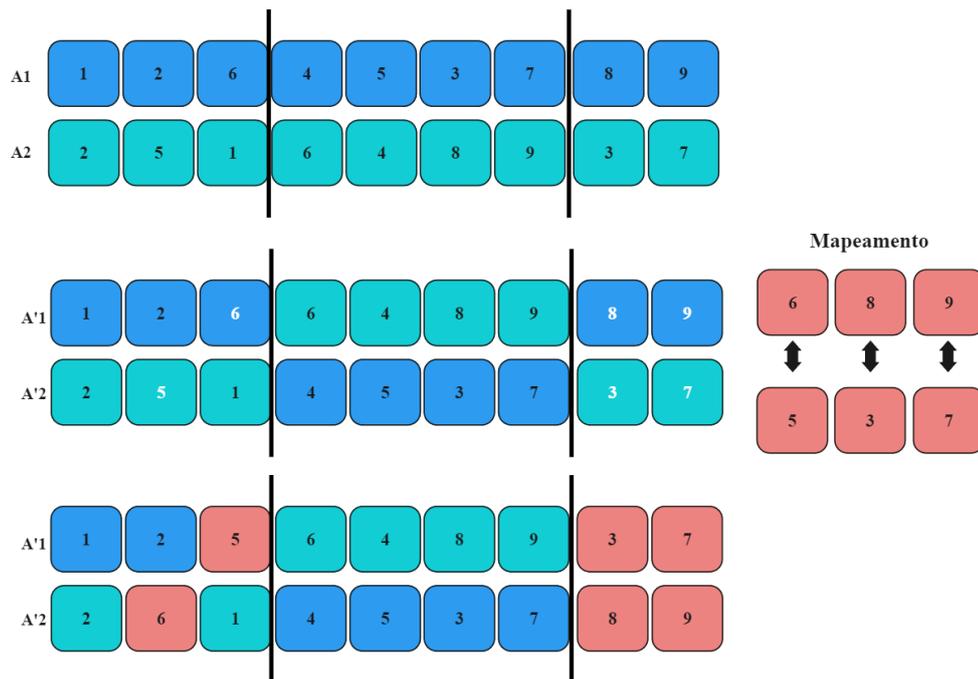
2.5.4 Cruzamento/Crossover

Para o problema de roteamento de veículos, vários operadores de cruzamento especializados são utilizados, pois os indivíduos criados pelos cruzamentos devem continuar sendo soluções válidas para o problema.

Um cruzamento é o parcialmente mapeado (PMX) (S; JOSEPH; JAYAKUMAR, 2015; WANG et al., 2020), onde os índices na parte do meio são trocados e as posições restantes são preenchidas com os genes do outro pai que ainda não estão presentes no filho, mantendo as suas posições originais sempre que possível. A Figura 11 mostra o funcionamento do

PMX, onde $A1$ e $A2$ são os pais e os pontos de corte são representados pelas linhas. No filho $A'1$ é replicado o segmento que está entre os pontos de corte do pai $A2$ que é $[6, 4, 8, 9]$, o restante dos genes são herdados conforme estão presentes no pai $A1$, analogamente esse processo é realizado no filho $A'2$. Posteriormente, são verificados os genes que são iguais nos filho $A'1$ e $A'2$, esses elementos iguais são mapeados e trocados pelos que estão ausentes mantendo as suas posições originais sempre que possível.

Figura 11: Cruzamento PMX.

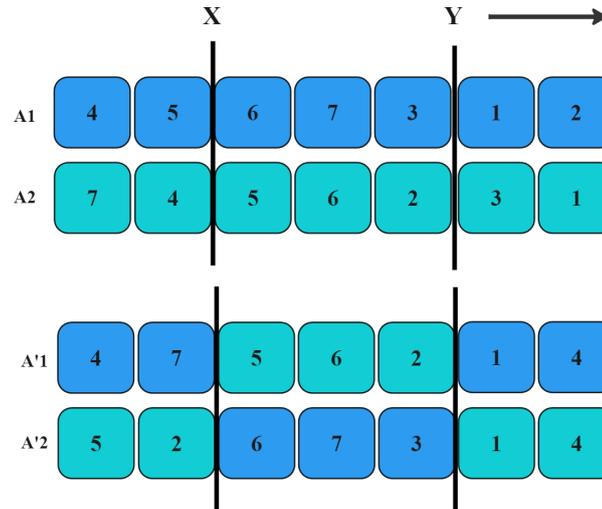


Fonte: Elaborada pelo autor.

O cruzamento ordenado (OX) (WANG et al., 2016; SINGH; L.; PUNDIR, 2019; HAERANI et al., 2017; LI; LI; PARDALOS, 2014) é um dos cruzamentos mais utilizados, onde a parte do meio (do ponto x ao ponto y) vem de um pai e o segundo pai é varrido do ponto $y + 1$ circularmente para completar o filho. A Figura 12 apresenta a operação do cruzamento OX, onde se tem os pais $A1$, $A2$ e o ponto de corte X e Y . No filho $A'1$ é replicado o segmento do pai $A2$ que é $[5, 6, 2]$. Posteriormente, com o objetivo de completar os genes do filho $A'1$, é copiado sequencialmente do pai $A1$ a partir de $y+1$, ignorando os genes que já estão no segmento copiado do pai $A2$. Analogamente, o processo é replicado em $A'2$, onde o segmento replicado é o do pai $A'1$ e o preenchimento dos genes restantes é copiado sequencialmente do pai $A2$.

2.5.5 Mutação

Figura 12: Cruzamento OX.



Fonte: Elaborada pelo autor.

Em relação à mutação aplicada no MDVRP, as mais utilizadas são a mutação de inserção (WANG et al., 2016) que remove o cliente de seu lugar e o insere em um local aleatório na rota em outra posição, o método de inversão, onde uma parte do cromossomo é invertida (S; JOSEPH; JAYAKUMAR, 2015; WANG et al., 2016) e a de troca, onde é realizada a troca de clientes de uma rota (OPOKU-AMANKWAAH; OMBUKI-BERMAN, 2017; WANG et al., 2016; MAHMUD; HAQUE, 2019; HAERANI et al., 2017; BOUYAHYIOUY; BELLABDAOUI, 2016).

2.6 Considerações

Compreender sobre as principais características, restrições e funcionalidades que compõem um problema de roteamento de veículos, é importante para o entendimento de como diversas atividades no cotidiano enfrentam esse problema, onde é de grande importância o esforço para obter melhores soluções para a essa categoria de problemas.

O referencial teórico abordou técnicas para solução do MDVRP onde foi apresentado as vantagens e o funcionamento dos algoritmos genéticos. A escolha do Algoritmo Genético para resolver o MDVRP, no trabalho proposto, deve-se à sua eficácia em encontrar soluções quase ótimas e à sua capacidade de lidar com as múltiplas restrições do problema. O próximo capítulo aborda o algoritmo genético posposto utilizado no presente estudo.

3 Material e Métodos

Este capítulo trata sobre o método aplicado para alcançar os objetivos propostos por este trabalho e é organizado conforme descrito a seguir. A representação do problema usando algoritmos genéticos é descrita na Seção 3.1. Na Seção 3.2 é retratado o método de clusterização desenvolvido para o estudo. A geração de população inicial é apresentada na Seção 3.3. Na Seção 3.4 é discutido a construção da função objetivo e a regra de penalização aplicada. O método de Elitismo é descrito na Seção 3.5. A Seção 3.6 apresenta o método de seleção utilizado. O cruzamento aplicado é descrito na Seção 3.7. A Seção 3.8 demonstra o operador de mutação. O método de Ejeção de Cliente é apresentado na Seção 3.9. Na Seção 3.10 é descrito o operador 2-opt. A Seção 3.11 apresenta o método 3-opt. O método de Combinações de Rotas é explicado na Seção 3.12. Por fim, a Seção 3.13 mostra o fluxograma do Algoritmo Genético Swap.

3.1 Representação do Problema usando Algoritmos Genéticos

Como discutido no Capítulo 2, os Algoritmos Genéticos, do inglês, *Genetic Algorithms* (GAs) são uma escolha popular para resolver o problema de roteamento de veículos com múltiplos depósitos devido à sua capacidade de encontrar soluções aceitáveis em um tempo razoável (BAI; GAO, 2010; GAO; JIA; LIU, 2016; TAUFIK; RAMADHAN; NABABAN, 2015; ZHANG et al., 2019). Os Algoritmos Genéticos em conjunto com algoritmos de busca local, são especialmente eficazes em problemas de otimização combinatorial, como o problema de roteamento de veículos com múltiplos depósitos, pois podem lidar com grandes conjuntos de soluções possíveis. Além disso, eles podem ser facilmente adaptados para lidar com várias restrições, como a capacidade dos veículos e as distâncias entre os clientes e depósitos (AZADEH; FARROKHI-ASL, 2017; OUDOUAR; FALLAHI; ZAOUI, 2019).

No Capítulo 2 foi apresentado a definição matemática do modelo MDVRP como um grafo $G = (V, E)$, onde V é o conjunto de nós, que representam os clientes e os depósitos, E é o conjunto de arestas que conectam cada par de nós que representam, por exemplo, a

distância entre eles. Além da definição do grafo, existem algumas restrições no MDVRP, como o número de veículos em cada depósito, a capacidade dos veículos, o tamanho das rotas e que cada cliente só é atendido uma única vez.

Uma solução do MDVRP pode ser representada da seguinte forma: dado um conjunto de n depósitos $D = \{d_1, d_2, \dots, d_n\}$ e um conjunto de m clientes $C = \{c_1, c_2, \dots, c_m\}$, uma solução para o MDVRP é um conjunto de rotas compostas por tuplas que inicial e terminam em um depósito d_i , onde $i = 1, 2, \dots, n$, e que contem uma sequência de clientes c_j , onde $j = 1, 2, \dots, m$. Como descrito anteriormente, cada cliente só é atendido uma única vez, ou seja, cada cliente só pode aparecer em uma tupla do conjunto de rotas. Uma possível solução para um problema contendo $D = \{d_1, d_2\}$ e $C = \{c_1, c_2, c_3, c_4, c_5\}$, poderia ser o conjunto de rotas $s = \{[d_1, c_3, c_1, d_1], [d_2, c_2, c_5, d_2], [d_2, c_4, d_2]\}$, onde a resposta do problema é dada pela soma das distâncias calculadas em cada rota.

Como um indivíduo em um GA representa uma solução para o problema, uma possível representação seria utilizar o conjunto de rotas s definido anteriormente, formando uma única tupla $s' = [d_1, c_3, c_1, d_1, d_2, c_2, c_5, d_2, d_2, c_4, d_2]$, onde cada cromossomo teria um identificador para representar o depósito ou cliente.

Neste trabalho, para simplificar a representação foram utilizados 2 conceitos. O primeiro foi a utilização de uma estrutura denominada cliente/depósito que relaciona os clientes com os depósitos, onde cada cliente é alocado a um único depósito. O segundo conceito é o de termino de uma rota em um depósito, onde os primeiros clientes vão sendo adicionados a uma rota até que atinja a capacidade do veículo, neste ponto uma nova rota é criada. Uma representação do conjunto de rotas s é uma tupla $s'' = [c_3, c_1, c_2, c_5, c_4]$, onde as duas primeiras posições da tupla são dos clientes do depósito d_1 e as demais do depósito d_2 .

Além da representação, a estrutura cliente/depósito define um padrão de funcionamento do algoritmo, onde os operadores do GA serão aplicados de forma separada em cada depósito (somente nos clientes do depósito) e a *fitness* ou aptidão, será a composição das rotas criadas em todos os depósitos. Como as operações do GA são executadas em cada depósito de forma separada, o GA poderia ser desmembrado em vários GA's, onde cada GA resolveria um problema VRP, e a solução final do MDVRP seria a composição da rota final dada pelo melhor resultado obtido em cada GA.

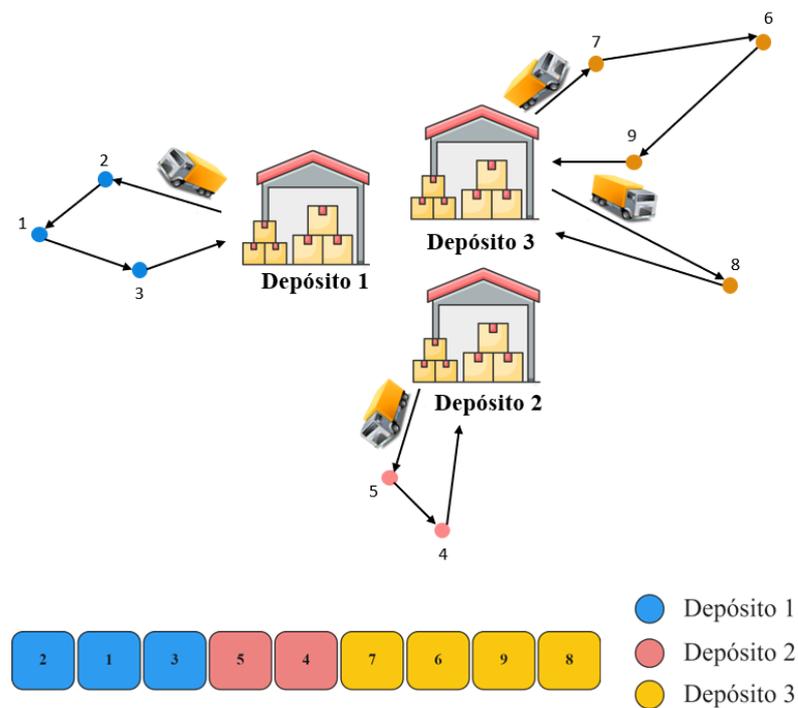
A definição da estrutura cliente/depósito afeta o resultado final do algoritmo, pois limita o espaço de busca, e definir a estrutura ideal é um problema difícil, devido à complexidade do problema. Uma forma de modificar a região de busca, durante a execução

do GA, será utilizando um operador de mutação entre depósitos que irá modificar a estrutura cliente/depósito e será discutido mais a frente.

Assim, a representação de cada indivíduo do GA corresponde a um vetor com uma sequência de rotas para atender os clientes pertencentes a cada depósito do MDVRP. Todas as posições do vetor são preenchidas com o identificador atribuído ao cliente, em que será realizada uma parada na rota. O conjunto dessas paradas representa uma rota do problema, respeitando a capacidade do veículo e a estrutura cliente/depósito. Cada depósito pode possuir uma ou mais rotas para atender todos seus clientes, desde que respeite a restrição de número de veículos disponíveis para atender o problema.

A Figura 13 ilustra uma representação do problema. A rota que atende os clientes [2, 1 e 3] pertence ao Depósito 1, a rota que possui os clientes [5 e 4] são do Depósito 2 e as duas últimas rotas [7, 6 e 9] e [8], separadas pelo critério de capacidade do veículo, são pertencentes ao Depósito 3. Um indivíduo do algoritmo genético para a estrutura cliente/depósito definida, seria a junção de todas as rotas, resultando no indivíduo [2, 1, 3, 5, 4, 7, 6, 9 e 8].

Figura 13: Representação do Problema.



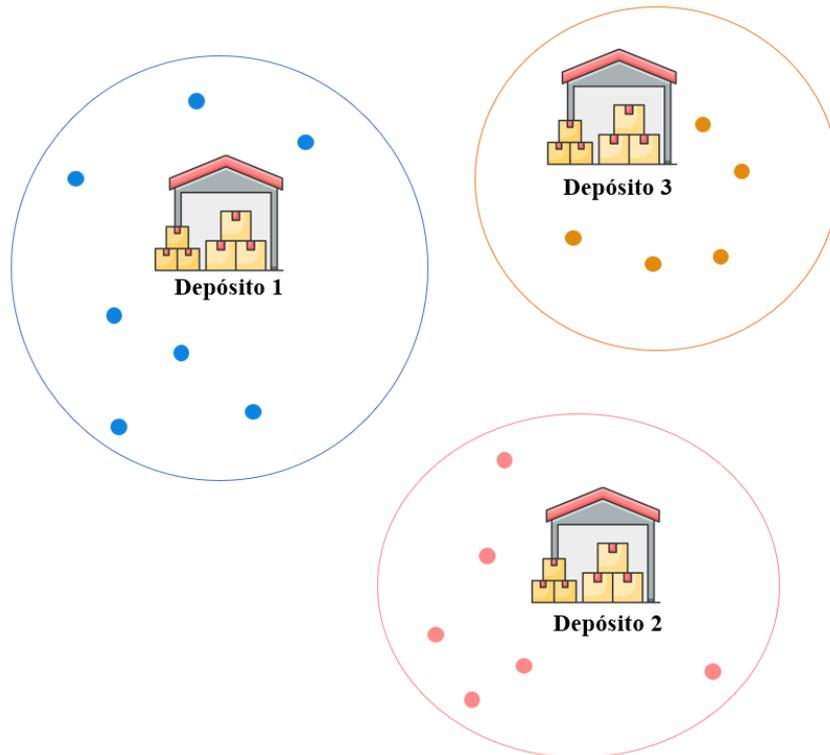
Fonte: Elaborada pelo autor.

3.2 Clusterização

No problema de MDVRP tem-se vários clientes e vários depósitos, e uma forma intuitiva de definir a estrutura cliente/depósito é utilizando a clusterização, pois clientes próximos de um depósito, tem tendência a ser atendido pelos veículos daquele depósito. Assim, inicializar o GA com uma população inicial que leva em consideração esta informação, pode reduzir o espaço inicial de busca e levar o algoritmo rapidamente para boas soluções.

A tarefa de clusterização consiste na tentativa de reunir elementos que possuem determinada semelhança entre si, de acordo com alguma restrição imposta pelo criador deste agrupamento. A Figura 14 indica a clusterização por distância entre clientes e depósitos, baseado em suas posições geográficas, onde os clientes são agrupados ao depósito mais próximo.

Figura 14: Clusterização entre clientes e depósitos, baseado em suas posições geográficas.



Fonte: Elaborada pelo autor.

Baseado na clusterização por distância, foi desenvolvido um método de clusterização, denominado Clusterização Swap, que utiliza um conjunto de listas de clientes que podem ser permutados entre os depósitos. A ideia é verificar se a troca de alguns clientes permitem reduzir o número de rotas necessárias nos depósitos. O intuito dessa operação é modificar

a estrutura cliente/depósito criada pela clusterização por distância, com o objetivo de aprimorar o espaço de busca inicial do algoritmo genético.

Para construir as listas de trocas, primeiramente é necessário definir a estrutura cliente/depósito baseado na clusterização por distância. Em seguida, é construído uma lista de troca para cada depósito que é formada por clientes, que não estão alocadas no depósito, mas que estão dentro de uma distância máxima permitida para cada cliente. Por fim é feito uma análise para remover da lista de troca de cada depósito, todos os clientes que estejam “distantes” dos clientes alocados no depósito. O Algoritmo 1 descreve o passo-a-passo para obter as listas de trocas entre depósitos, onde o algoritmo recebe como entrada um conjunto de clientes C , um conjunto de depósitos D e uma parâmetro p que será utilizado para definir a distância máxima permitida para cada cliente pertencer à lista de troca.

Algoritmo 1: MONTA AS LISTAS DE TROCAS ENTRE DEPÓSITOS

```

Entrada:  $C, D, p$ ; //  $C$  Conjunto de clientes,  $D$  conjunto de depósitos,  $p$  parâmetro
           distância
Saída:  $CD, L$ ; //  $CD$  a estrutura cliente/depósito.  $L$  listas de trocas entre
           depósitos
1 início
2   para todo  $c \in C$  faça
3     para todo  $d \in D$  faça
4        $M[c][d] \leftarrow \text{distanciaEuclidiana}(d, c)$ ;
5     fim
6      $(dmin[c], k) \leftarrow \text{menorDistancia}(M[c], c)$ ;
7      $dmax[c] \leftarrow dmin[c] + dmin[c] * p$ ;
8      $\text{insereLista}(c, CD[k])$ ; //  $c$  é alocado ao depósito mais perto  $k$ 
9   fim
10  para todo  $d \in D$  faça
11    para todo  $c \in C$  que não está em  $CD[d]$  faça
12       $dist \leftarrow \text{distanciaEuclidiana}(d, c)$ ;
13      se  $dist \leq dmax[c]$  então
14         $\text{insereLista}(c, L[d])$ ;
15      fim
16    fim
17  fim
18  /* remover clientes distantes da lista L */
19  para todo  $d \in D$  faça
20    para todo  $c \in L[d]$  faça
21       $(distCD, k) \leftarrow \text{menorDistanciaCD}(CD[d], d, c)$ ;
22      se  $distCD > dmin[k]$  então
23         $\text{removeLista}(c, L[d])$ ;
24      fim
25    fim
26  fim
27 retorna  $CD, L$ ;

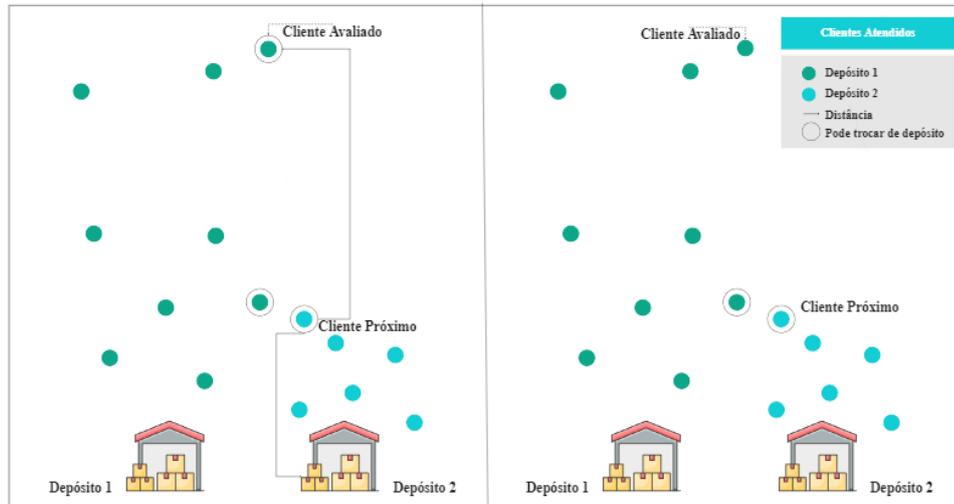
```

O Algoritmo 1 funciona da seguinte forma:

1. Uma matriz de adjacência M de distâncias entre os clientes e os depósitos, é criada utilizando a distância euclidiana entre eles (linhas 2 a 5).
2. A cada cliente c é definido qual depósito k está mais próximo e qual é a distância entre eles. Para isso, é utilizado a função *menorDistancia* que retorna, além de k , a distância entre eles que é denominada de $dmin$. Em seguida é calculado a distância $dmax$ que define a maior distância que o cliente pode ficar de qualquer depósito. A distância $dmax$ é calculada utilizando o parâmetro p , que é um valor utilizado para definir a porcentagem da distância de $dmin$ que é adicionada à $dmin$ para compor $dmax$. Os valores de $dmin$ e $dmax$ de cada cliente c são armazenados em estruturas com o mesmo nome (linhas 6 e 7).
3. A estrutura cliente/depósito CD é montada de forma gradual para cada cliente c processado, onde o depósito k que é o depósito mais próximo de c é inserido na lista $CD[k]$ indicando que o cliente c está alocado ao depósito k (linha 8).
4. Nas linhas de 10 à 17 é montada as listas de troca L de cada depósito d . Para isso, é selecionado um depósito d e verificado para cada cliente c do conjunto de clientes C , quais clientes não estão alocados no depósito d , ou seja, não estão na lista $CD[d]$ (linhas 10 e 11). Para cada cliente c que não está em $CD[d]$ é calculado a distância dele para o depósito d e armazenado em $dist$ (linha 12). Em seguida é verificado se $dist$ é menor ou igual a $dmax[c]$, ou seja, se o depósito d está dentro da distância máxima de c , caso seja verdade c é inserido na lista de troca do depósito d , neste caso em $L[d]$ (linhas 13 à 15).
5. Após a montagem das listas L é feita uma análise para remover os clientes que estão “distantes” dos clientes alocados no depósito. Para tal, é calculado para cada cliente c , alocado na lista de troca de um depósito $L[d]$, qual é a menor distância de c para qualquer cliente alocado em d , isto é, que estão na lista $CD[d]$. Para este calculo é utilizado a função *menorDistanciaCD* que retorna o depósito k mais próximo de c e o valor da distância entre eles em $distCD$. A seguir é verificado se $distCD$ é maior que a distância mínima de k com o depósito d que esta armazenado em $dmin[k]$, caso seja verdade, c é removido da lista de troca do depósito $L[d]$ (linhas 18 à 24). A Figura 15 ilustra um caso onde um cliente é removido da lista de troca.

A partir das listas de trocas dos depósitos é aplicado o método Clusterização Swap, que analisa as listas de trocas para verificar se é possível reduzir o número de rotas necessárias

Figura 15: Verificação de distância de clientes distantes.



Fonte: Elaborada pelo autor.

em um depósito enviando um cliente para outro depósito, sem aumentar o número de rotas do depósito que vai receber o cliente. O funcionamento do Clusterização Swap é definido no Algoritmo 2 que recebe como entrada o conjunto de depósitos D , as listas contendo as estruturas cliente/depósito CD , as listas de trocas L e a capacidade dos veículos cap .

O algoritmo começa calculando para cada depósito d a demanda total de clientes alocados usando a função *depositoDemanda* e atribui o valor para $demandaD[d]$ e então, ordena a lista de trocas entre depósitos $L[d]$ em ordem decrescente de demanda dos clientes usando a função *ordenaDemanda* (linhas 2 a 4). Em seguida, o algoritmo analisa para cada depósito d , se a inserção de um cliente c da lista de trocas $L[d]$, não aumenta o número mínimo de rotas necessárias para atender a todos os clientes. O número mínimo de rotas $nrotas_d$ é calculado dividindo a demanda total $demandaD[d]$ pela capacidade do veículo cap . Por fim, o algoritmo verifica se a retirada do cliente c , do depósito que ele está alocado k , reduz o número mínimo de rotas necessárias em k , ou seja, $nrotas_k$. Caso o número mínimo de rotas do depósito d permaneça o mesmo e o número mínimo de rotas do depósito k diminua, então o cliente c é removido do depósito k e alocado no depósito d , ou seja, removido da lista $CD[k]$ e alocado na lista $CD[d]$ (linhas 6 a 21).

3.3 População Inicial

Foi desenvolvido um método guloso para geração da população inicial, onde, após ser feita a clusterização em cada depósito, é selecionado um cliente aleatoriamente e criado uma rota que conecta o cliente ao depósito. A partir desse cliente, a rota começa a ser

Algoritmo 2: ALGORITMO CLUSTERIZAÇÃO SWAP

Entrada: D, CD, L, cap ; // D conjunto de depósitos, CD estruturas cliente/depósito; L listas de trocas entre depósitos, cap capacidade do veículo

Saída: CD, L

```

1 início
2   para todo  $d \in D$  faça
3      $demandaD[d] \leftarrow demandaDeposito(CD[d]);$ 
4      $L[d] \leftarrow ordenaDemanda(L[d]);$ 
5   fim
6   para todo  $d \in D$  faça
7     para todo  $c \in L[d]$  faça
8        $k \leftarrow obtemDeposito(c);$ 
9        $nrotas_d \leftarrow demandaD[d]/cap;$ 
10       $nrotas_k \leftarrow demandaD[k]/cap;$ 
11       $aux_d \leftarrow (demandaD[d] + demanda(c))/cap;$ 
12       $aux_k \leftarrow (demandaD[k] - demanda(c))/cap;$ 
13      se  $(nrotas_d == aux_d)$  e  $(nrotas_k > aux_d)$  então
14        removeLista( $c, L[d]$ );
15        removeLista( $c, CD[k]$ );
16        insereLista( $c, CD[d]$ );
17         $demandaD[d] = demandaD[d] + demanda[c];$ 
18         $demandaD[k] = demandaD[k] - demanda[c];$ 
19      fim
20    fim
21  fim
22 fim
23 retorna  $CD, L;$ 

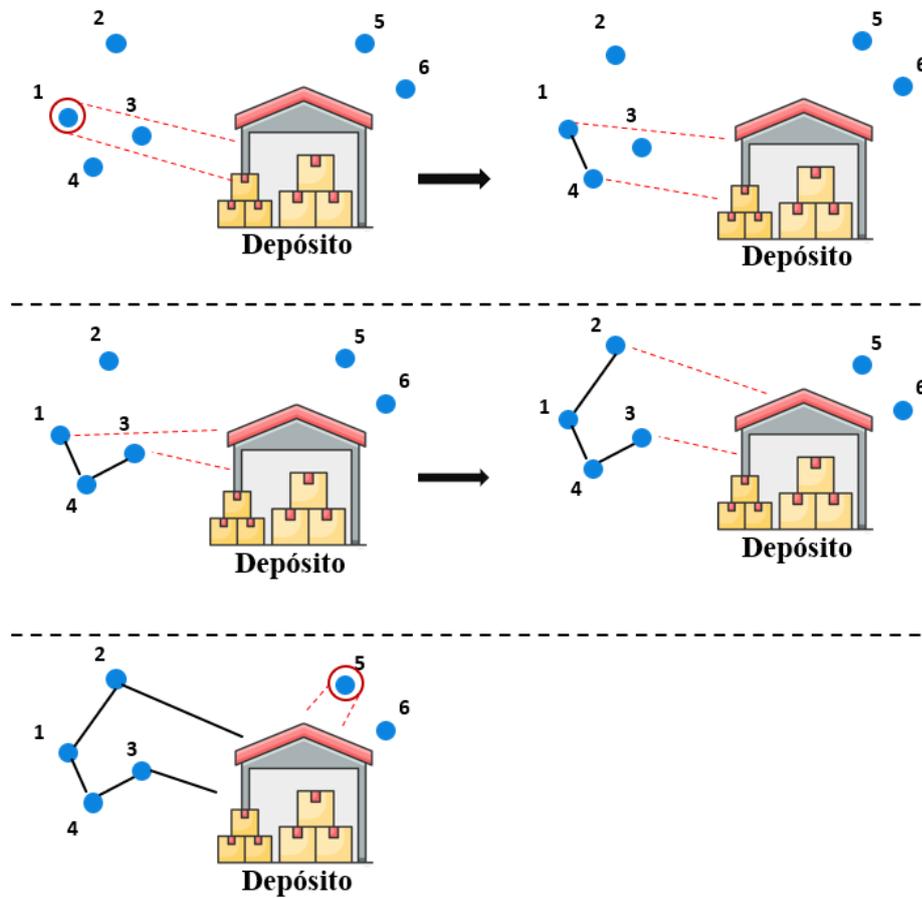
```

expandida, adicionando o cliente com a distância mais próxima a uma das extremidades. Assim, a cada inserção de um novo cliente, é verificado o cliente do início e do final da rota em construção, adicionado um cliente que ainda não faz parte de uma rota e esteja mais próximo do cliente do início ou do final

A Figura 16 demonstra o método de geração de rotas na população inicial. Inicialmente, será selecionado um cliente aleatório pertencente ao depósito, o cliente c_1 . Posteriormente, o cliente c_1 é conectado com o cliente mais próximo, o cliente c_4 . Logo após, são analisados os clientes mais próximo entre c_1 e c_4 . O cliente c_3 é o mais próximo de c_4 , comparado com os clientes disponíveis em relação ao cliente c_1 . Portanto, o cliente c_3 é conectado ao cliente c_4 . Depois é comparado qual é o cliente mais próximo entre c_1 ou c_3 . Como o cliente c_2 é mais próximo de c_1 , os dois são conectados. Em seguida, é verificado qual cliente é mais próximo dos clientes c_2 e c_3 , sendo c_5 o cliente mais próximo. Como o cliente c_5 estoura a capacidade do veículo, uma nova rota é criada para o cliente c_5 e posteriormente o cliente c_6 é adicionado à rota que contem o cliente c_5 resultando nas rotas $[3, 4, 1, 2]$ e $[5, 6]$.

Como explicado, a capacidade do veículo é utilizada para identificar o fim de uma

Figura 16: Método de geração de rotas da população inicial.



Fonte: Elaborada pelo autor.

rota, ou seja, quando o veículo não é capaz de atender o próximo cliente, o veículo retorna para o depósito de origem e inicia uma nova rota. O conjunto de rotas de todos depósitos representa um indivíduo pertencente a população do algoritmo genético, conforme descrito na Seção 3.1.

3.4 Função Objetivo

Através da função objetivo é possível realizar a avaliação dos indivíduos que compõem a população do GA. A função objetivo do MDVRP é definida como a somatória das distâncias de todas as rotas de todos os depósitos que compõem o indivíduo.

Como descrito na Seção 3.1, um indivíduo do GA pode ser representado por uma tupla de clientes divididas por uma estrutura cliente/depósito, onde os k_1 primeiros clientes pertencem ao depósito 1, os próximos k_2 clientes pertencem ao depósito 2 e assim por diante. Dentre os clientes de um depósito, o início e o término de cada rota é definido

pela capacidade do veículo, onde as rotas são lidas de forma sequencial em que os clientes aparecem na tupla, até que não seja mais possível adicionar o próximo cliente a ser atendido pelo veículo (Figura 13). Como existe uma quantidade máxima de veículos para cada depósito, neste trabalho, caso a última rota não consiga atender todos os clientes do depósito, a última rota conterà todos os clientes restantes e a função de objetivo será penalizada.

A penalização é uma técnica que atribui um custo adicional às soluções que violam as restrições impostas pelo problema, tornando-as menos atrativas para o processo de seleção e reprodução dos algoritmos genéticos. Essa penalização pode ser aplicada, por exemplo, aumentando o custo da função objetivo em problemas de minimização. A penalização de soluções, nos algoritmos genéticos, ajudam a garantir que as soluções, que atendam a todas as restrições do problema, possuam uma maior aptidão, permitindo que soluções inválidas, mesmo tendo uma menor aptidão, possam participar do processo de reprodução aumentando a diversidade da população.

Uma forma de calcular a função objetivo teria a seguinte sequência:

1. Dado um indivíduo $s = [c_3, c_1, c_2, c_5, c_4]$ do GA;
2. Converta o indivíduo s de forma a representar as rotas, respeitando a estrutura cliente/depósito e a capacidade do veículo. Cada rota tem como início e fim o depósito que ela está alocada. Uma possível conversão de s seria $s' = [[d_1, c_3, c_1, d_1], [d_2, c_2, c_5, d_2], [d_2, c_4, d_2]]$, onde s' possui 3 rotas;
3. Em seguida, percorra cada rota somando a distância de um ponto ao próximo da rota, esta somatória vai definir a distância de cada rota. Também verifica se a rota estourou a capacidade do veículo.
4. A solução da função objetivo S é dada pela Equação 3.1, onde S_{rotas} é o somatório das distâncias das rotas, calculadas no item anterior, nre é o número de rotas que estouraram a capacidade do veículo e P é uma variável binária que possui valor 1 (um) quando alguma rota estourou a capacidade e 0 (zero), caso contrário. A penalização aplicada leva em consideração o numero de rotas que estouraram a capacidade do veículo nre , pois quanto menor for o valor de nre , mais próximo de uma solução válida o indivíduo vai estar.

$$S = S_{rotas} + S_{rotas} \times P \times (0,3 + 0,2 \times nre) \quad (3.1)$$

Como a composição das rotas em um depósito é feita adicionando os clientes até que a demanda do próximo cliente ultrapasse a capacidade do veículos, uma outra forma de montar as rotas seria avaliar a sequência de clientes na ordem inversa, ou seja, do cliente no final da tupla até o cliente inicial. Uma possível formação para o exemplo anterior seria $s' = [[d_1, c_3, c_1, d_1], [d_2, c_2, d_2], [d_2, c_5, c_4, d_2]]$.

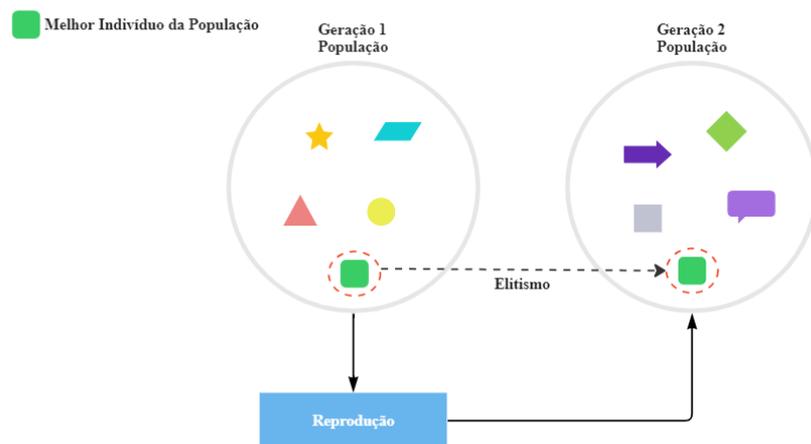
Desta forma, neste trabalho o cálculo da função objetivo é feito analisando a ordem sequencial e inversa do segmento de clientes organizados em um depósito, pois a forma que os clientes estão ordenados interfere no cálculo da aptidão do indivíduo, onde o melhor valor obtido representa a aptidão do indivíduo no GA.

3.5 Elitismo

O processo de elitismo é utilizado para evitar que soluções de alta qualidade sejam perdidas ao longo do processo de evolução, garantindo que essas soluções continuem a ser consideradas nas próximas gerações.

Para o presente trabalho foi definido a permanência do melhor indivíduo obtido em uma geração, garantindo que uma solução de qualidade não seja perdida no processo de evolução das próximas gerações. A Figura 17 apresenta o processo de elitismo aplicado no algoritmo genético desenvolvido.

Figura 17: Elitismo no Algoritmo Genético.



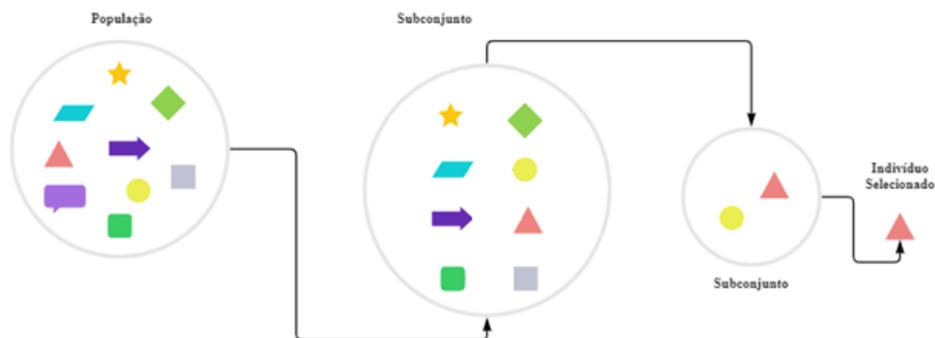
Fonte: Elaborada pelo autor.

3.6 Seleção

A seleção consiste na escolha de indivíduos com base em sua aptidão, para acasalarem e produzirem descendentes. A seleção de torneio consiste em selecionar aleatoriamente n indivíduos de um grupo e a partir da aptidão de ambos é selecionado o indivíduo vencedor (MAHMUD; HAQUE, 2019; OPOKU-AMANKWAAH; OMBUKI-BERMAN, 2017; OUDOUAR; FALLAHI; ZAOUI, 2019). Para determinar um vencedor é utilizado um coeficiente que diz a probabilidade do melhor indivíduo vencer o torneio, ou seja, se o coeficiente for 0,75 o melhor indivíduo terá 75% de chance de vencer o indivíduo com menor aptidão.

O processo de seleção neste trabalho é desenvolvido em duas fases. Na primeira fase é criado um subconjunto da população, onde são selecionados, a cada iteração, dois indivíduos de forma aleatória e aplicado o processo de torneio até o preenchimento do subconjunto. Finalmente, na segunda fase, é selecionado dois indivíduos no subconjunto e aplicado o processo de torneio, resultando na aquisição do indivíduo vencedor. A operação genética de torneio é representada pela Figura 18.

Figura 18: Seleção Torneio.



Fonte: Elaborada pelo autor.

3.7 Cruzamento

O operador genético de cruzamento desempenha um papel fundamental no processo de busca e geração de soluções, pois promove a diversidade genética, explorando diferentes combinações de genes e possibilitando a descoberta de soluções mais eficientes e adaptáveis. Esse operador permite a combinação de informações genéticas de dois pais para criar descendentes com características únicas. Nessa seção é descrita a utilização do operador

genético de cruzamento conhecido como *Order Crossover* (OX) utilizado em (AZADEH; FARROKHI-ASL, 2017; HAERANI et al., 2017; LI; LI; PARDALOS, 2014; RABBANI et al., 2018).

Dado dois pais $A1$ e $A2$, o algoritmo genético seleciona um ponto inicial e um ponto final em cada pai, que definirá a sequência de genes que serão herdadas pelos filhos. O segmento selecionado do pai $A1$ é replicado no filho $A'2$ e o segmento selecionado do pai $A2$ é copiado no filho $A'1$ na mesma posição dos pais. Essa seleção de pontos permite que os filhos herdem uma parte específica da solução dos pais.

Uma vez que os segmentos são copiados para os filhos, o restante dos genes é preenchido utilizando os genes do pai que não forneceu o segmento, ou seja, o filho $A'1$ é preenchido com os genes do pai $A1$ e o filho $A'2$ é preenchido com os genes do pai $A2$. Os genes são preenchidos saindo do início dos genes do pai que irá fornecer as características restantes, seguindo em ordem sequencial, copiando os genes que não corresponda à parte herdada anteriormente. Essa abordagem garante diversidade genética e a introdução de novas combinações, contribuindo para a busca de soluções ótimas.

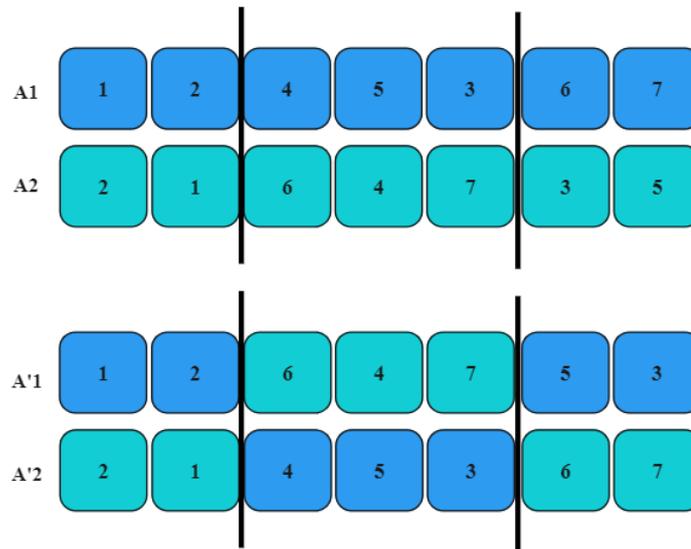
O operador genético criado é exemplificado pela Figura 19, onde se tem os pais $A1$ e $A2$ e o ponto de corte, definido de forma aleatória, a partir do índice 2 e finalizando no índice 4. No filho $A'1$ é replicado o segmento do pai $A2$ que é [6, 4 e 7]. Posteriormente, com o objetivo de completar os genes do filho $A'1$, é copiado sequencialmente do pai $A1$ a partir do gene inicial, ignorando os genes que já estão no segmento copiado do pai $A2$. Analogamente, o processo é replicado em $A'2$, onde o segmento replicado é o do pai $A'1$ e o preenchimento dos genes restantes é copiado sequencialmente do pai $A2$.

A Figura 20 ilustra o método do cruzamento aplicado ao problema de roteamento de veículos com múltiplos depósitos, onde o processo descrito anteriormente é aplicado em cada depósito, respeitando a estrutura cliente/depósito, resultando uma solução diversificada e conseqüentemente permitindo encontrar soluções melhores na evolução genética.

3.8 Mutação

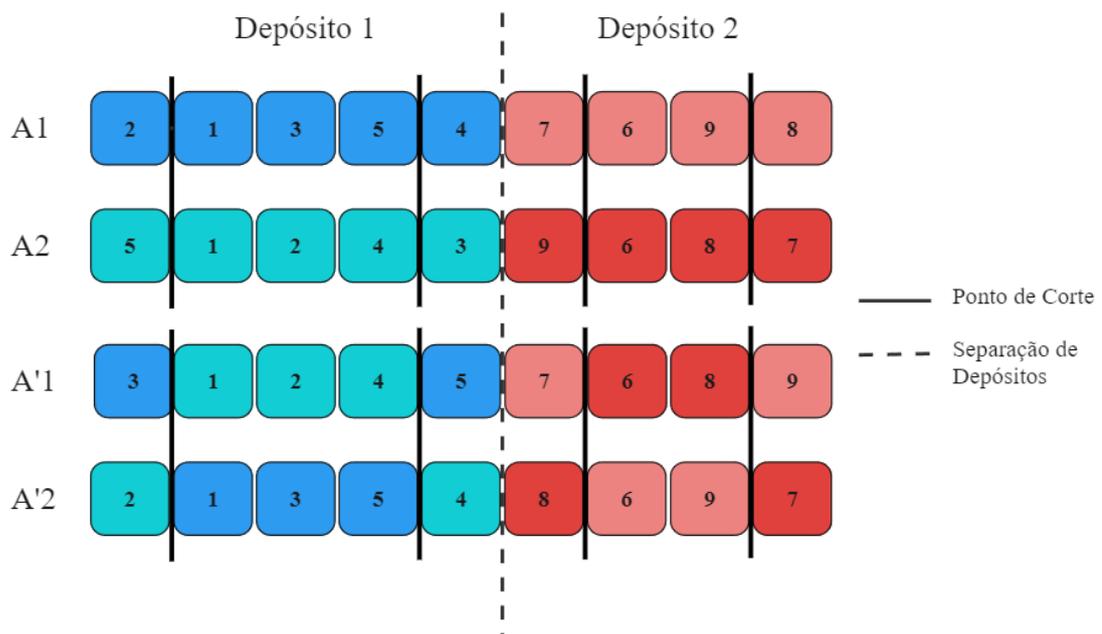
A mutação é um operador que tem o objetivo de ampliar um pouco o espaço de busca. Da mesma forma que acontece na natureza, a mutação tem pouca chance de acontecer e o mesmo comportamento ocorre no Algoritmo Genético. No GA é definido geralmente uma pequena probabilidade para ocorrer a mutação, com o intuito de não perder todas as características dos pais.

Figura 19: Cruzamento.



Fonte: Elaborada pelo autor.

Figura 20: Cruzamento Múltiplos Depósitos.



Fonte: Elaborada pelo autor.

O presente estudo realiza dois métodos de mutação. O primeiro, conhecido como método de inversão, é executado dentro da estrutura cliente/depósito de cada depósito, realizando alterações em suas rotas. O outro método, denominado de mutação entre depósitos, modifica a estrutura cliente/depósito dos indivíduos, onde um cliente é removido de um depósito e inserido em outro.

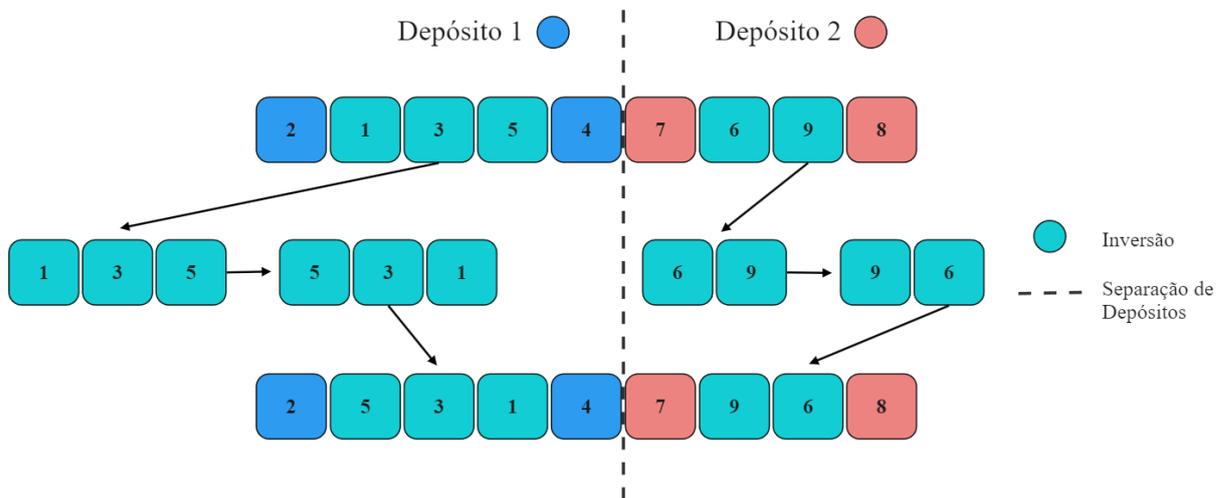
O método de inversão é utilizado para alterar as rotas dentro de cada depósito, ele

inverte uma parte do cromossomo com o objetivo de melhorar a produção de soluções (OUDOUAR; FALLAHI; ZAOUI, 2019; WANG et al., 2016). Inicialmente, em cada segmento do cromossomo que corresponde as rotas de cada depósito, é observado se irá ocorrer a mutação, de acordo com a taxa de mutação. Ocorrendo a mutação, são escolhidos dois pontos aleatórios que designam o início e o fim da parte do cromossomo que irá ser invertida e finalmente é aplicada a mutação.

A operação de mutação é exemplificada pela Figura 21. No Depósito 1, os clientes atendidos estão na seguinte ordem para serem atendidos [2, 1, 3, 5, 4]. A mutação seleciona dois pontos para realizar a inversão, o ponto inicial é o cliente 1 e o final é o cliente 5. Portanto o segmento [1, 3, 5] será invertido, resultado [5, 3, 1]. Por fim, a disposições dos clientes no depósito 1 após a mutação fica [2, 5, 3, 1, 4].

Como no Depósito 1, o Depósito 2 também foi selecionado para a mutação. Inicialmente os clientes atendidos são [7, 6, 9, 8]. A mutação acontece e os clientes 6 e 9 são os pontos inicial e final para a operação de inversão, resultado na troca de posição dos dois pontos. O resultado da ordem dos clientes atendidos no Depósito 2 ficou da seguinte forma [7, 9, 6, 8].

Figura 21: Mutação de inversão.



Fonte: Elaborada pelo autor.

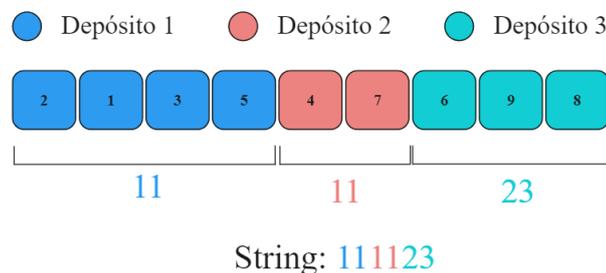
A mutação entre depósitos utiliza uma lista de clientes que podem ser permutados entre os depósitos, como utilizado em (MAHMUD; HAQUE, 2019; OMBUKI-BERMAN; HANSHAR, 2009). Neste trabalho a lista de troca utilizada é a mesma criada no algoritmo Clusterização Swap apresentado na Seção 3.2. A mutação entre depósitos seleciona de forma aleatório um depósito d e, em seguida, de forma aleatória é escolhido um cliente c da lista de trocas do depósito d . Por fim, o cliente c é removido do seu depósito de origem

e inserido em d , modificando a estrutura cliente/depósito dos indivíduos da população.

Com o intuito de evitar que um cliente saia do depósito e retorne para seu depósito de origem, foi criada uma memória de curto prazo de tamanho n , que armazena o estado atual da estrutura cliente/depósito, das últimas n mutações, em uma *string*. A *string* consiste na soma de todos os identificadores dos clientes que estão contidos em cada depósito, sendo realizada a concatenação de todos esses valores. Quando a memória está cheia, o estado mais antigo é removido da memória de curto prazo.

A Figura 22 ilustra a criação de uma *string* que seria armazenada na memória de curto prazo. O Depósito 1 contém os clientes na seguinte ordem de atendimento [2, 1, 3 e 5] a soma desses identificadores retorna o valor 11. No Depósito 2 os clientes atendidos são o [4 e 7], a soma dos identificadores resulta em 11. Por fim, no depósito 3 os clientes atendidos são [6, 9 e 8], a soma dos identificadores dos clientes é 23. A *string* gerada é a concatenação das somas dos identificadores dos clientes de todos os depósitos, resultando na *string* 111123 que configura o estado atual das disposições dos clientes nos depósitos.

Figura 22: String memória.



Fonte: Elaborada pelo autor.

Após a mutação entre depósitos, toda a população é reconstruída para a nova estrutura cliente/depósito. Neste trabalho, 80% da população atual o cliente é inserido logo após o cliente mais próximo no depósito em que foi inserido. E os 20% restantes da população, são reconstruídas as rotas utilizando o método de geração de rotas exemplificado na Seção 3.3. Importante destacar que a melhor solução encontrada no GA e sua estrutura cliente/depósito são armazenadas antes da mutação entre depósitos, até que uma solução melhor a substitua.

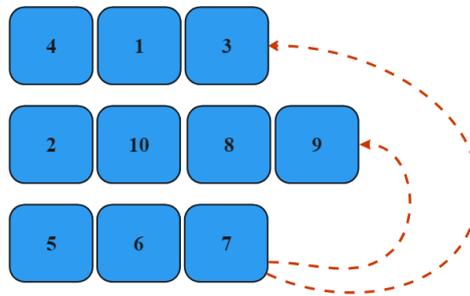
3.9 Ejeção de Cliente

A ejeção de cliente foi desenvolvida com o objetivo de melhorar a solução obtida. Esse método consiste em uma sequência de movimentos coordenados de clientes de uma

rota para uma rota sucessiva. Cada ejeção envolve k níveis (rotas) iniciando da rota k e terminando na rota 1. Ele tenta ejetar o cliente da posição n da rota k , ultimo cliente, para a rota $k - 1$, caso não seja possível, isto é, violou alguma restrição do problema, ele tenta ejetar de forma sucessiva o cliente na rota $k - 2$, $k - 3$ até atingir a rota 1. Caso o cliente na posição n não possa ser ejetado, tenta-se inserir o cliente $n - 1$, o cliente $n - 2$ até que uma ejeção seja possível ou esgote todos os clientes da rota k (TALBI, 2009).

A Figura 23 exemplifica o processo de Ejeção de Cliente, com a tentativa de ejetar o cliente 7 em outras rotas, respeitando as restrições impostas pelo problema.

Figura 23: Ejeção de Cliente.



Fonte: Elaborada pelo autor.

3.10 2-opt

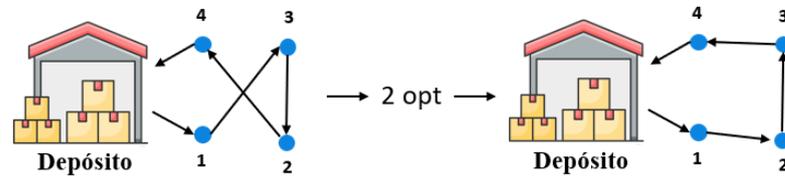
A operação 2-opt é um algoritmo de busca local que tem o objetivo de melhorar a qualidade das soluções encontradas, realizando trocas entre pares de elementos em uma solução existente para buscar uma solução mais promissora.

O algoritmo 2-opt encontra rotas que possuam caminhos que se cruzam e busca reinserir o cliente de modo que não ocorra cruzamento de rotas (KARAKATIĆ; PODGORELEC, 2015). O 2-opt remove duas arestas de um caminho previamente gerado e reconecta os dois novos caminhos criados. A reconexão é feita de forma a manter o caminho válido e a operação é executada apenas quando o novo caminho for mais curto do que o antigo. O processo do 2-opt continua até que nenhuma melhoria adicional seja possível (DAVENDRA, 2010).

A Figura 24 ilustra a operação 2-opt, onde a rota inicial é a $[1, 3, 2, 4]$ que possui um cruzamento que provavelmente aumenta o tamanho da rota mínima possível. O algoritmo 2-opt analisa a rota realizando inversões das disposição dos clientes, com o objetivo de reconectar os caminhos e reduzir o tamanho das rotas. Essas trocas de posições são feitas

até que não exista mais melhora na rota. O resultado final da rota foi $[1, 2, 3, 4]$ eliminando o cruzamento que existia anteriormente.

Figura 24: Operação 2-opt.



Fonte: Elaborada pelo autor.

3.11 3-opt

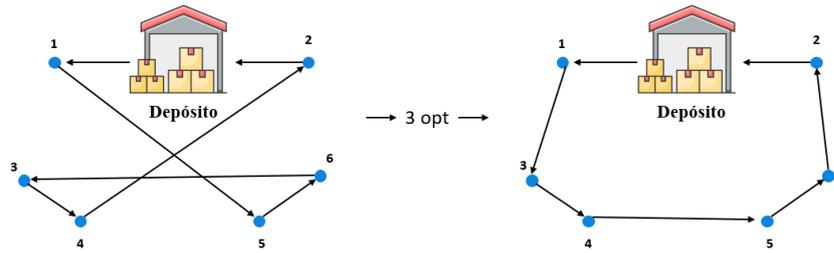
A operação 3-opt é uma extensão da operação 2-opt. Assim como a operação 2-opt, a 3-opt tem o objetivo de melhorar a qualidade das soluções encontradas pelo algoritmo, realizando trocas entre elementos em uma solução existente. No entanto, a operação 3-opt vai além, permitindo trocas entre conjuntos de três elementos, o que pode levar a melhorias significativas em problemas complexos.

O 3-opt envolve a identificação de três conjuntos de elementos não adjacentes em uma solução existente e a reorganização desses conjuntos de forma a buscar uma solução mais promissora. A pesquisa é concluída quando não há mais movimentos de 3-opt que possam melhorar a qualidade do caminho (DAVENDRA, 2010).

A principal vantagem da operação 3-opt é a sua capacidade de explorar um espaço de busca maior. Ao permitir trocas entre três conjuntos de elementos, ela é capaz de encontrar rotas alternativas que não seriam alcançadas apenas com a operação 2-opt. Essa capacidade de explorar diferentes configurações pode levar a melhorias significativas na qualidade da solução encontrada pelo algoritmo genético, mas exige mais tempo de processamento do que o 2-opt.

A operação de 3-opt é exemplificada na Figura 25 a rota inicial é representada pelo segmento $[1, 5, 6, 3, 4, 2]$. Como o 2-opt, o 3-opt analisa a rota realizando inversões das disposição dos clientes, com o objetivo de reconectar os caminhos e reduzir o tamanho das rotas. Essas trocas de posições são feitas até que não exista mais melhora na rota. O resultado após a operação foi $[1, 3, 4, 5, 6, 2]$ eliminando os cruzamentos que existia anteriormente.

Figura 25: Operação 3-opt.



Fonte: Elaborada pelo autor.

3.12 Combinações de Rotas

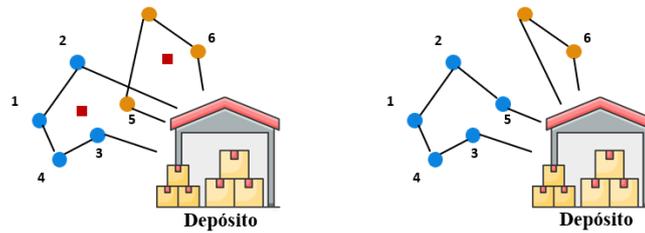
Com o objetivo de evitar cruzamento de duas rotas diferentes e otimizar a disposição das rotas, foi desenvolvido um método que tenta combinar clientes de duas rotas e otimizar a solução do problema final essa operação foi denominada de combinação de rotas.

O método de combinação de rotas funciona da seguinte maneira. Inicialmente calcula-se o ponto médio de todos os clientes, de cada rota pertencente a um depósito. Posteriormente, é criada uma matriz de distância que contém a distância entre cada ponto médio das rotas. Logo após, através da matriz de distância tenta-se combinar cada rota com a mais próxima. A operação de combinação é feita comparando uma rota denominada R com a rota mais próxima R_{prox} . Caso exista algum cliente em R_{prox} que está mais próximo do ponto médio de R este cliente é retirado da rota R_{prox} e inseridos na rota R . Os clientes são inseridos na rota após ao cliente mais próximo.

Um exemplo de combinação de rotas é apresentada na Figura 26, onde se tem duas rotas. A primeira rota chamada de R_1 contém a seguinte ordem de atendimento de clientes [3, 4, 1, 2], a rota R_2 atende os clientes [5, 7, 6]. O método desenvolvido verifica os pontos médios das duas rotas, o ponto médio é representado por um quadrado na figura. Verifica-se a distância de cada cliente da rota de R_2 com o ponto médio das duas rotas. Por fim, os clientes de R_2 com a distância menor com o ponto médio de R_1 comparado com o de R_2 , são removidos de R_2 e adicionados R_1 . No exemplo, o cliente 5 é removido de R_2 e adicionado logo após ao cliente mais próximo em R_1 . O resultado final das rotas após a execução do método de combinação de rotas para R_1 resulta em [3, 4, 1, 2, 5] e R_2 foi produzido a rota [7, 6].

Após ser feita a combinação de rotas é aplicado a operação de 2-opt, com o objetivo de eliminar possíveis cruzamentos em rotas de um depósito, que podem ter sido criados, após a manipulação de clientes.

Figura 26: Combinação de rotas.



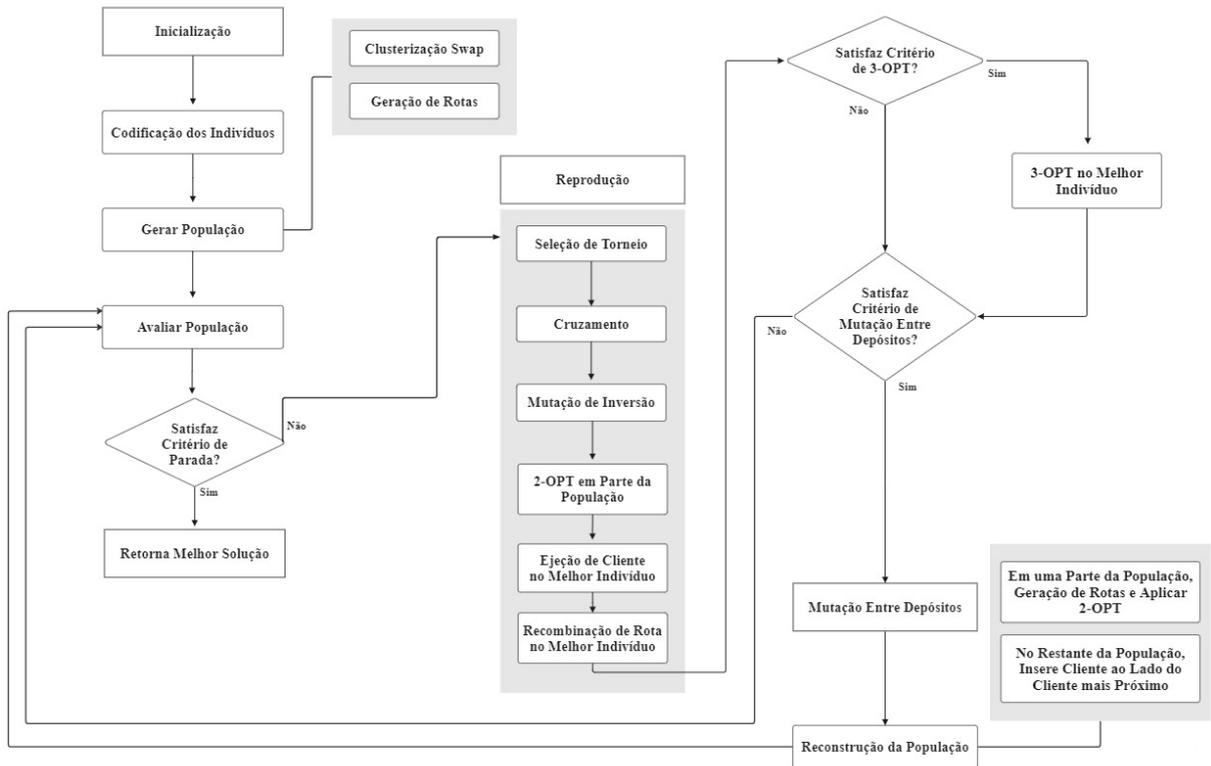
Fonte: Elaborada pelo autor.

3.13 Fluxograma Algoritmo Genético Swap

A Figura 27 apresenta o fluxo de funcionamento do Algoritmo Genético implementado no presente estudo. O algoritmo genético se inicia com a codificação dos indivíduos, apresentado na Seção 3.1. Posteriormente é formado uma população inicial, onde a estrutura cliente/depósito é definida pela a Custerização Swap e a geração de rotas de um indivíduo é formada usando o método guloso, descrito nas Seções 3.2 e 3.3. Logo após, é utilizada a função de aptidão onde são avaliados os indivíduos da população apresentada na Seção 3.4 e é feito o processo de elitismo, que consiste em selecionar o melhor indivíduo da geração, apresentado na Seção 3.5. Posteriormente, realiza-se o processo de reprodução, seleção de torneio, cruzamento e mutação de inversão, descritos nas Seções 3.6, 3.7 e 3.8 respectivamente. São utilizados os operadores 2-opt, ejeção de cliente com o objetivo de melhoria dos indivíduos da população, fazendo o remanejamento dos clientes nas rotas dos depósitos, apresentados nas Seções 3.10 e 3.9. Com o intuito de melhorar o melhor indivíduo da população, é aplicado o método de Recombinação de rotas, descrito na Seção 3.12. O próximo passo, o GA verifica se vai ser aplicado a execução do 3-opt no melhor indivíduo da população, apresentado na Seção 3.11. Como o 3-opt demanda muito tempo de processamento, ele não é executado em todas as gerações. Finalmente, é verificado se será realizada a mutação entre depósitos, mostrada na Seção 3.8. A mutação entre depósito modifica a estrutura cliente/depósito, ou seja, modifica a região do espaço de busca e, por isso, deve ser aplicado de acordo com algum critério como o número de iterações, número de iterações sem melhoras na solução ou outros. Todo o processo descrito da avaliação da população até a mutação entre depósitos é repetida até que satisfaça o critério de parada do Algoritmo Genético.

A seguir, o próximo capítulo apresenta os experimentos e resultados deste trabalho.

Figura 27: Fluxograma do algoritmo genético desenvolvido.



Fonte: Elaborada pelo autor.

4 Experimentos e Resultados

Este capítulo trata sobre os experimentos e os resultados gerados de acordo com os objetivos propostos para a realização desse trabalho e está estruturado conforme descrito a seguir. A descrição da base de dados é apresentada na Seção 4.1. Na Seção 4.2 são apresentados os experimentos e as discussões dos resultados.

4.1 Descrição da Base de Dados

As instâncias da base de dados são referentes a problemas de roteamento de veículos com múltiplos depósitos utilizados por (CORDEAU; GENDREAU; LAPORTE, 1997). Foram selecionadas 11 instâncias que possuem somente a restrição de capacidade e número de veículos como características do problema.

A Tabela 2 apresenta informações sobre as instâncias utilizadas no trabalho, sendo que a primeira coluna indica o nome da instância, a partir da segunda coluna até a quarta apresenta informações sobre as variáveis do problema e a última coluna mostra o melhor resultado encontrado pela literatura.

4.2 Experimentos

O algoritmo genético desenvolvido foi executado em quatro experimentos, o primeiro não permitindo a mutação de clientes entre depósitos e os demais permitindo a mutação. Como os Algoritmos Genéticos ou *Genetic Algorithms* (GAs) são métodos estocásticos, cada configuração do GA foi executada por 30 vezes em cada experimento e foi utilizado o número máximo de gerações como critério de parada.

Em relação aos resultados, foram feitas análises para os experimentos, comparando os valores obtidos referentes a porcentagem do desvio do melhor resultado conhecido na literatura (*gap*), média e desvio padrão. O *gap* é definido na Equação 4.1 (WANG et al.,

Tabela 2: Descrição das instâncias da base de dados.

Instância	N° de Depósitos	N° de Clientes	N° de Veículos	Capacidade	Melhor Resultado
p01	4	50	4	80	576,87
p02	4	50	2	160	473,53
p03	5	75	3	140	641,19
p04	2	100	8	100	1001,59
p05	2	100	5	200	750,03
p06	3	100	6	100	876,5
p07	4	100	4	100	885,8
p12	2	80	6	60	1318,95
p15	4	160	5	60	2505,42
p18	6	240	5	60	3702,85
p21	9	360	5	60	5474,84

Fonte: CORDEAU; GENDREAU; LAPORTE (1997).

2016).

$$gap = \frac{Sol. - M.Sol.}{M.Sol.} \times 100\% \quad (4.1)$$

Durante a implementação do GA foram considerados diferentes parâmetros de forma de avaliar a performance do Algoritmo Genético em diferentes configurações. Após vários testes empíricos, foi definido o conjunto de parâmetros do GA que foram aplicados em todos os experimentos. O primeiro parâmetro escolhido foi o tamanho da população de 150 indivíduos. No processo de reprodução é utilizado, na seleção dos indivíduos, o tamanho do subconjunto é igual a 10% da população e o melhor indivíduo possui 75% de chance para vencer o torneio, a taxa de cruzamento de 70% e a taxa de mutação de inversão de 2%. Após a reprodução dos novos indivíduos, o algoritmo 2-opt é aplicado em 10% da população que é selecionada de forma aleatória. Por fim, no processo de reprodução é aplicado a ejeção de cliente e a recombinação de rotas no melhor indivíduo. Em relação à mutação entre depósitos, ela é executada pela primeira vez na geração 1000 e depois a cada 200 gerações. Foi escolhido o tamanho 5 para a memória de curto prazo, que é responsável por evitar que um cliente saia do depósito e retorne para seu depósito de origem. Na reconstrução da população, após ser aplicado a mutação entre depósitos, é gerado 20% de indivíduos utilizando o método guloso e aplicado o 2-opt nestes indivíduos, os outros 80% são atualizados inserindo o cliente que foi trocado de depósito ao lado do cliente mais próximo. Os parâmetros estão compilados na Tabela 3.

Durante a execução do GA, o algoritmo pode atingir um mínimo local e não conseguir

Tabela 3: Parâmetros utilizados no algoritmo genético desenvolvido

Parâmetro	Valor
Tamanho da População	150
Tamanho do subconjunto da seleção de torneio	10% da população
Seleção de torneio	75%
Taxa de cruzamento	70%
Taxa de mutação de inversão	0,2%
2-opt	10% da população
Mutação entre depósitos (primeira)	1000 geração
Mutação entre depósitos (próximas)	a cada 200 gerações
Memória de curto prazo	tamanho 5
Reconstrução da população (atualizados)	80% da população
Reconstrução da população (novos)	20% da população

Fonte: Elaborada pelo autor.

evoluir a população por algumas gerações. Para tentar aumentar a diversidade da população ampliando a região de busca do algoritmo, também foi implementado o seguinte procedimento. Durante as iterações, caso o algoritmo não obtenha um indivíduo melhor durante 49 gerações consecutivas é aplicado o algoritmo 3-opt no melhor indivíduo. Se na próxima geração não surgir um indivíduo melhor, a taxa de mutação é dobrada e a contagem é reiniciada. Este procedimento se repete até que a taxa de mutação atinja o valor máximo de 2% ou seja obtido um indivíduo melhor, neste caso, a taxa de mutação é restaurada ao valor inicial.

Também é importante destacar que a melhor solução encontrada no GA e sua estrutura cliente/depósito, antes da mutação entre depósitos, é armazenada até que uma solução melhor a substitua.

4.2.1 Experimento 1

O objetivo do experimento foi avaliar o impacto da estrutura cliente/depósito definido na população inicial do GA, portanto não foi permitido a mutação que realiza a troca de clientes entre depósitos. Foram utilizadas 3 estruturas cliente/depósito (CD), sendo elas:

- **CD_D** baseado na clusterização por distância;
- **CD_S20** baseado no algoritmo de Clusterização Swap com a distância de 20%;
- **CD_S40** baseado no algoritmo de Clusterização Swap com a distância de 40%;

Em relação à escolha das distâncias de 20% e 40% para o algoritmo de Clusterização

Swap, elas foram feitas de forma a obter estruturas cliente/depósito diferentes da clusterização de distância, mas sem permitir que clientes muito distantes possam ser trocados entre os depósitos.

A Tabela 4 apresenta a quantidade de clientes diferentes de CD_S20 e CD_S40 ao serem comparadas com CD_D. Observa-se que a mudança da porcentagem de distância permitida para a troca de clientes entre depósitos, gera um impacto na geração da população inicial. Conseqüentemente, esse impacto interfere na geração dos espaços de buscas de soluções, realizadas na evolução do algoritmo genético. Entretanto, observa-se que para as instâncias p04, p12, p15, p18 e p21 não houve mudança na estrutura cliente/depósito. Esse comportamento nos diz que para essas instâncias o GA desenvolvido apresentará soluções próximas ao CD_D. O motivo do CD_S ou algoritmo de Clusterização Swap, ter esse comportamento é por ser uma técnica que parte de CD_D com ajustes que buscam uma alteração na estrutura cliente/depósito. Como a estrutura continuou igual é como se o GA fosse executado para CD_D.

Tabela 4: Clientes diferentes de CD_S20 e CD_S40 comparados com CD_D.

	CD_S20	CD_S40
Instância	Diferentes	Diferentes
p01	0	2
p02	0	1
p03	1	3
p04	0	0
p05	1	1
p06	2	1
p07	3	3
p12	0	0
p15	0	0
p18	0	0
p21	0	0

Fonte: Elaborada pelo autor.

Neste experimento o número máximo de gerações foi de 3000. A Tabela 5 apresentada as soluções e o *gap* para cada instância da literatura, onde os melhores valores estão destacados em negrito.

Analisando o resultado de CD_S20, CD_S40 com CD_D para as instâncias p04, p12, p15, p18 e p21 os resultados encontrados foram próximos ou idênticos. Como discutido anteriormente, todas tem a mesma estrutura cliente/depósito, como foi apresentado na Tabela 4. Para avaliar o impacto da Clusterização Swap, foi construída a Tabela 6 que apresenta os resultados para as instâncias com a estrutura cliente/depósito diferente.

Tabela 5: AG sem mutação de trocas de clientes entre depósitos com distância de clusterização ajustada para 20%.

Inst.	M.Sol	CD_S20		CD_S40		CD_D	
		Solução	Gap	Solução	Gap	Solução	Gap
p01	576,87	612,53	0,0618	631,48	0,0947	612,53	0,0618
p02	473,53	507,01	0,0707	498,95	0,0537	507,01	0,0707
p03	641,19	685,85	0,0696	689,00	0,0746	688,44	0,0737
p04	1001,59	1051,47	0,0498	1060,34	0,0587	1049,92	0,0483
p05	750,03	805,15	0,0735	786,55	0,0487	803,14	0,0708
p06	876,5	918,10	0,0475	918,77	0,0482	919,31	0,0488
p07	885,8	938,92	0,0600	930,21	0,0501	924,51	0,0437
p12	1318,95	1365,69	0,0354	1365,69	0,0354	1365,69	0,0354
p15	2505,42	2731,37	0,0902	2731,37	0,0902	2731,37	0,0902
p18	3702,85	4097,06	0,1065	4097,06	0,1065	4097,06	0,1065
p21	5474,84	6209,66	0,1342	6203,16	0,1330	6207,23	0,1338

Fonte: Elaborada pelo autor.

Tabela 6: Comparação dos resultados sem mutação entre depósitos das estruturas CD_20, CD_S40 e CD_D.

Inst.	M.Sol	CD_S20		CD_S40		CD_D	
		Solução	Gap	Solução	Gap	Solução	Gap
p01	576,87	612,53	0,0618	631,48	0,0947	612,53	0,0618
p02	473,53	507,01	0,0707	498,95	0,0537	507,01	0,0707
p03	641,19	685,85	0,0696	689,00	0,0746	688,44	0,0737
p05	750,03	805,15	0,0735	786,55	0,0487	803,14	0,0708
p06	876,5	918,10	0,0475	918,77	0,0482	919,31	0,0488
p07	885,8	938,92	0,0600	930,21	0,0501	924,51	0,0437

Fonte: Elaborada pelo autor.

Os resultados da Tabela 6 mostram que a estrutura CD_S20, comparado com a CD_D, apresentou dois resultados melhores, dois resultados piores e dois resultados iguais, onde os resultados iguais das instâncias p01 e p02 saem da mesma estrutura cliente/depósito. Comparando CD_S40 com CD_D verifica-se que cada estrutura foi melhor em 3 das 6 instâncias. Este resultado mostra que o algoritmo Clusterização Swap pode encontrar estruturas cliente/depósito mais promissoras que a clusterização por distância.

Os valores de média e o desvio padrão (DP) das 30 execuções de cada configuração são apresentados na Tabela 7. Analisando o desvio padrão, observa-se que todos as configurações apresentam um desvio padrão pequeno. Este resultado garante que o GA entregará resultados próximos da média apresentada.

Tabela 7: Média e desvio padrão do GA sem mutação de trocas de clientes entre depósitos.

Inst.	M. Sol	CD_S20		CD_S40		CD_D	
		Média	DP	Média	DP	Média	DP
p01	576,87	620,88	5,84	632,55	1,93	622,94	5,62
p02	473,53	510,74	4,56	501,92	3,58	511,12	5,34
p03	641,19	692,17	3,75	693,95	2,40	694,76	3,64
p04	1001,59	1086,73	18,51	1080,63	11,65	1082,07	17,91
p05	750,03	826,85	16,87	813,28	11,13	833,65	16,23
p06	876,5	948,22	14,10	944,59	15,26	942,34	13,34
p07	885,8	968,12	15,03	961,04	15,66	939,83	9,64
p12	1318,95	1382,15	20,23	1380,78	19,05	1382,91	18,14
p15	2505,42	2758,84	32,20	2768,75	28,96	2758,09	26,05
p18	3702,85	4170,33	41,96	4167,68	39,96	4168,96	44,96
p21	5474,84	6335,20	67,88	6328,79	70,95	6334,60	78,57

Fonte: Elaborada pelo autor.

Analisando o resultados das médias de CD_S20, CD_S40 com CD_D para as instâncias p04, p12, p15, p18 e p21, verifica-se que cada estrutura apresentou 2 melhores resultados. Comparando CD_S20 com CD_D, a estrutura CD_S20 obteve a melhor média em 4 instâncias e comparando CD_S40 com CD_D, cada estrutura apresentou a melhor média em 3 instâncias.

Tabela 8: Comparação das médias e desvio padrão sem mutação entre depósitos das estruturas CD_20, CD_S40 e CD_D.

Inst	M. Sol.	CD_S20		CD_S40		CD_D	
		Média	DP	Média	DP	Média	DP
p01	576,87	620,88	5,84	632,55	1,93	622,94	5,62
p02	473,53	510,74	4,56	501,92	3,58	511,12	5,34
p03	641,19	692,17	3,75	693,95	2,40	694,76	3,64
p05	750,03	826,85	16,87	813,28	11,13	833,65	16,23
p06	876,5	948,22	14,10	944,59	15,26	942,34	13,34
p07	885,8	968,12	15,03	961,04	15,66	939,83	9,64

Fonte: Elaborada pelo autor.

Como obter uma boa estrutura cliente/depósito, para um problema qualquer é complexo, uma forma de ampliar a região de busca do GA mudando a estrutura cliente/depósito é permitindo a mutação entre depósitos. Os próximos experimentos exploram esta possibilidade.

4.2.2 Experimento 2

No segundo experimento a mutação de clientes entre depósitos foi utilizada. O objetivo foi avaliar o impacto da mutação de clientes entre depósitos no GA. A estrutura cliente/depósito utilizada foi a CD_S20 utilizada no Experimento 1, sendo executada 3000 gerações.

A Tabela 9 apresenta o resultado da execução do GA com e sem a mutação de clientes entre depósitos. Os resultados indicam que a mutação de clientes entre depósitos obteve melhores resultados do que a execução do GA sem a aplicação desse método. A mutação trouxe ganhos na obtenção de melhores soluções, permitindo que o GA explorasse novos espaços de busca e que seus indivíduos conseguissem evoluir para soluções melhores.

Tabela 9: Comparação de GA executado com e sem mutação de clientes entre depósitos

Inst.	M. Sol.	CD_S20			
		Sem Mutação		Com Mutação	
		Solução	gap	Solução	gap
p01	576,87	612,53	0,0618	606,68	0,0517
p02	473,53	507,01	0,0707	482,36	0,0186
p03	641,19	685,85	0,0696	663,56	0,0349
p04	1001,59	1051,47	0,0498	1049,20	0,0475
p05	750,03	805,15	0,0735	806,82	0,0757
p06	876,5	918,10	0,0475	917,18	0,0464
p07	885,8	938,92	0,0600	927,15	0,0467
p12	1318,95	1365,69	0,0354	1343,71	0,0188
p15	2505,42	2731,37	0,0902	2681,78	0,0704
p18	3702,85	4097,06	0,1065	4131,81	0,1158
p21	5474,84	6209,66	0,1342	6166,05	0,1263

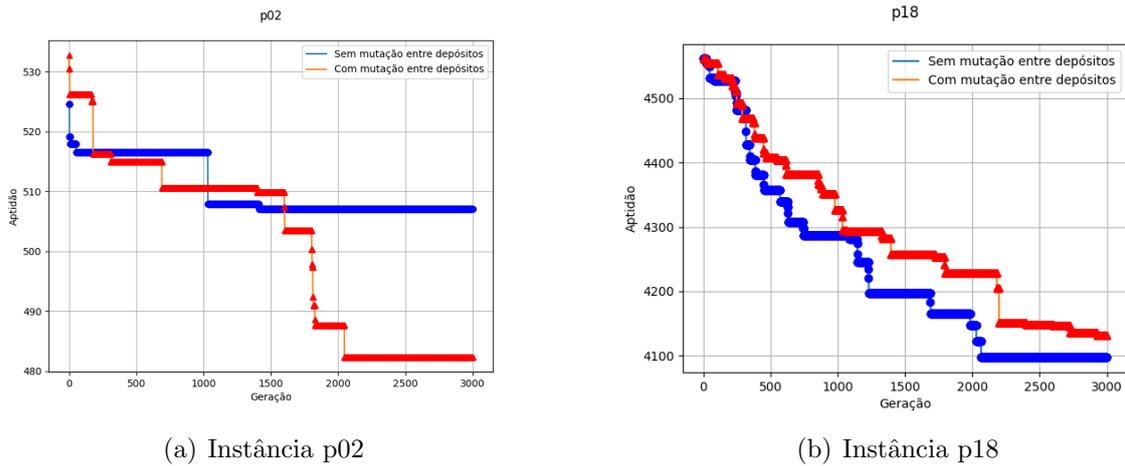
Fonte: Elaborada pelo autor.

A Figura 28 (a) apresenta uma execução da instância p02, onde são apresentadas em cada linha os valores de aptidão do melhor indivíduo encontrado até o momento em cada geração. Observa-se que existe uma certa convergência entre os dois algoritmos até a geração 1000, onde até o momento não foi ativada a mutação de clientes entre depósitos. Logo após a milésima geração, o algoritmo genético que permite a mutação de clientes entre depósitos, melhora a solução obtida. Entretanto, o GA sem essa mutação fica estagnado.

Na Figura 28 (b) é apresentado o gráfico para a instância p18, os indivíduos evoluem com o passar das gerações e o algoritmo que não possui a mutação de clientes entre depósitos fica estagnado acima de 2000 gerações. O algoritmo que realiza a mutação,

apresenta um comportamento de evolução constante com o passar das gerações. A linha que representa esse algoritmo, mostra que os indivíduos poderiam continuar evoluindo, caso não fosse alcançado o critério de parada do GA.

Figura 28: Algoritmo genético sem mutação de clientes entre depósitos comparado com Algoritmo genético com mutação de clientes entre depósitos



Fonte: Elaborada pelo autor.

4.2.3 Experimento 3

O terceiro experimento teve como objetivo avaliar as listas de trocas entre clientes proposta neste trabalho com as listas de trocas propostas em (MAHMUD; HAQUE, 2019; OMBUKI-BERMAN; HANSHAR, 2009).

Foram utilizadas 4 estruturas cliente/depósito (CD), sendo elas:

- **CD_S20** baseado no algoritmo Clusterização Swap com a distância de 20%;
- **CD_S50** baseado no algoritmo Clusterização Swap com a distância de 50%;
- **CD_A20** baseado no algoritmo Clusterização Swap utilizando as listas de trocas propostas em (MAHMUD; HAQUE, 2019; OMBUKI-BERMAN; HANSHAR, 2009), com a distância de 20%;
- **CD_A50** baseado no algoritmo Clusterização Swap utilizando as listas de trocas propostas em (MAHMUD; HAQUE, 2019; OMBUKI-BERMAN; HANSHAR, 2009), com a distância de 50%;

A principal diferença da abordagem utilizada em (MAHMUD; HAQUE, 2019; OMBUKI-BERMAN; HANSHAR, 2009) comparadas com a desse trabalho, é que as listas de troca entre

clientes desenvolvidas não permitem que um cliente candidato que esteja distante dos clientes alocados no depósito, pertença a lista deste depósito, garantindo a proximidade dos clientes atendidos.

A escolha das distâncias de 20% e 50%, foram com o intuito de observar o impacto que o aumento da distância máxima permitida para troca de clientes geram nos dois tipos de listas avaliadas. Com o propósito de verificar a evolução do algoritmo em um período maior, possibilitando uma maior quantidade de mutações de troca de clientes entre depósitos, neste experimento foram utilizadas 5000 gerações.

A Tabela 10 indicam os resultados para as estruturas cliente/depósito analisadas. Os resultados demonstram que os algoritmos CD_S foram melhores que o CD_A. O CD_S20 apresentou melhores resultados na maioria das instâncias e o CD_S50 foi melhor em 3 instâncias. Comparando o CD_S20 e o CD_A20, ou seja, as listas com distâncias de 20%, nota-se que o CD_S20 obteve melhores respostas em 8 das 11 instâncias, empatou em duas e perdeu somente na p01. Em relação às listas com distância de 50%, o CD_S50 obteve melhores respostas em todas as instâncias ao ser comparado com CD_A50. Este resultado mostra que a remoção de clientes distantes proposta neste trabalho melhorou o desempenho do algoritmo ao ser comparado com a lista proposta em (MAHMUD; HAQUE, 2019; OMBUKI-BERMAN; HANSHAR, 2009). Analisando o comportamento do ajuste de distância de 20% e 50%, nota-se que o ajuste impacta na evolução dos resultados obtidos pelo algoritmo genético. No qual a distância ajustada em 20% apresenta melhores resultados que a de 50% nas instâncias estudadas.

Tabela 10: Melhores resultados do GA com mutação entre depósitos com distância de troca permitida ajustada em 20% e 50%.

Inst	M. Sol.	CD_S20		CD_S50		CD_A20		CD_A50	
		Solução	Gap	Solução	Gap	Solução	Gap	Solução	Gap
p01	576,87	606,68	0,0517	603,22	0,0457	606,32	0,0511	620,82	0,0762
p02	473,53	482,36	0,0186	481,18	0,0162	482,36	0,0186	532,98	0,1256
p03	641,19	663,56	0,0349	669,81	0,0446	686,01	0,0699	678,28	0,0578
p04	1001,59	1049,20	0,0475	1051,90	0,0502	1051,54	0,0499	1089,37	0,0876
p05	750,03	793,46	0,0579	797,01	0,0626	798,92	0,0652	863,47	0,1512
p06	876,5	917,18	0,0464	928,13	0,0589	920,08	0,0497	989,77	0,1292
p07	885,8	927,15	0,0467	946,74	0,0688	927,83	0,0475	1023,59	0,1556
p12	1318,95	1341,84	0,0174	1320,74	0,0014	1341,84	0,0174	1326,85	0,0060
p15	2505,42	2681,78	0,0704	2714,76	0,0836	2684,48	0,0715	2855,03	0,1395
p18	3702,85	4055,56	0,0953	4080,44	0,1020	4075,51	0,1006	4343,48	0,1730
p21	5474,84	6118,34	0,1175	6195,60	0,1316	6124,06	0,1186	6679,97	0,2201

Fonte: Elaborada pelo autor.

Os resultados dos valores da média e desvio padrão das 30 execuções das instâncias

dos algoritmos CD_S e CD_A ajustados com a distância 20% e 50% são apresentados na Tabela 11. Os resultados da média e desvio padrão indicam que CD_S20 e CD_A20 obtiveram melhores resultados em suas execuções, comparadas ao CD_S50 e CD_A50. Isso ressalta como o ajuste da distância de troca de clientes entre depósitos possui um impacto significativo sobre a evolução do algoritmo genético na obtenção dos resultados. Comparando pelas distâncias, o CD_S20 obteve melhores médias em 7 das 11 instâncias ao ser comparado com CD_A20 e o CD_S50 obteve melhores respostas em 10 das 11 instâncias ao ser comparado com CD_A50. Analisando o desvio padrão, observa-se que todos as configurações apresentam um desvio padrão pequeno garantindo que o GA sempre entregará resultados próximos da média apresentada.

Tabela 11: Média e desvio padrão do GA com mutação entre depósitos com distância de troca permitida ajustada em 20% e 50%.

Inst	M. Sol.	CD_S20		CD_S50		CD_A20		CD_A50	
		Média	DP	Média	DP	Média	DP	Média	DP
p01	576,87	613,33	3,74	625,99	9,84	612,79	3,05	663,48	14,51
p02	473,53	488,23	4,95	502,14	8,02	492,42	6,19	536,93	2,35
p03	641,19	677,37	7,09	691,47	7,52	696,17	5,23	701,49	10,91
p04	1001,59	1075,47	11,69	1082,12	17,77	1079,00	9,30	1122,44	22,67
p05	750,03	818,88	11,84	825,97	12,94	817,64	10,30	896,28	13,10
p06	876,5	945,75	12,43	951,36	13,80	944,84	15,62	1025,76	17,47
p07	885,8	952,25	10,54	969,96	17,09	956,57	19,50	1110,53	29,63
p12	1318,95	1354,03	15,31	1369,18	21,93	1354,20	15,94	1364,29	26,88
p15	2505,42	2737,20	32,17	2762,26	23,66	2725,70	32,62	2960,37	47,93
p18	3702,85	4121,69	37,22	4185,65	51,78	4124,01	35,86	4499,19	62,08
p21	5474,84	6253,12	51,80	6342,59	86,90	6259,84	66,13	6866,09	67,31

Fonte: Elaborada pelo autor.

4.2.4 Experimento 4

O quarto e último experimento teve o objetivo de avaliar e comparar as estruturas cliente/depósito utilizando a mutação entre depósitos com as listas de trocas desenvolvidas neste trabalho.

Foram utilizadas as seguintes estruturas cliente/depósito (CD):

- **CD_D20** baseado na clusterização por distância com as listas de trocas com distância de 20%;
- **CD_S20** baseado no algoritmo Clusterização Swap com a distância de 20%.

Na Tabela 12 os resultados de CD_S20 e CD_D20 indicam que o CD_S20 obteve melhores respostas em 6 das 11 instâncias, empatou em duas e perdeu em três. Além disso, obteve o melhor resultado na instância p04 ao ser comparado com a estrutura CD_20, mostrando que o algoritmo Clusterização Swap produziu pequenas melhoras em relação à clusterização por distância.

Tabela 12: Melhores resultados do GA com mutação de trocas de clientes entre depósitos.

Inst.	M. Sol.	CD_S20		CD_D20	
		Solução	Gap	Solução	Gap
p01	576,87	606,68	0,0517	608,61	0,0550
p02	473,53	482,36	0,0186	482,36	0,0186
p03	641,19	663,56	0,0349	664,92	0,0370
p04	1001,59	1049,20	0,0475	1058,00	0,0563
p05	750,03	793,46	0,0579	797,30	0,0630
p06	876,5	917,18	0,0464	911,99	0,0405
p07	885,8	927,15	0,0467	915,47	0,0335
p12	1318,95	1341,84	0,0174	1341,84	0,0174
p15	2505,42	2681,78	0,0704	2706,26	0,0802
p18	3702,85	4055,56	0,0953	4049,45	0,0936
p21	5474,84	6118,34	0,1175	6134,01	0,1204

Fonte: Elaborada pelo autor.

Tabela 13: Média e desvio padrão do GA com mutação de trocas de clientes entre depósitos.

Inst.	M. Sol.	CD_S20		CD_D20	
		Média	DP	Média	DP
p01	576,87	613,333	3,743	613,591	3,053
p02	473,53	488,234	4,948	490,730	6,742
p03	641,19	677,366	7,090	678,850	7,640
p04	1001,59	1075,470	11,685	1075,295	10,653
p05	750,03	818,875	11,840	826,207	15,484
p06	876,5	945,754	12,433	942,786	14,391
p07	885,8	952,249	10,544	940,701	9,448
p12	1318,95	1354,030	15,310	1359,213	16,215
p15	2505,42	2737,203	32,165	2741,377	21,530
p18	3702,85	4121,685	37,217	4120,160	36,989
p21	5474,84	6253,124	51,801	6236,780	63,965

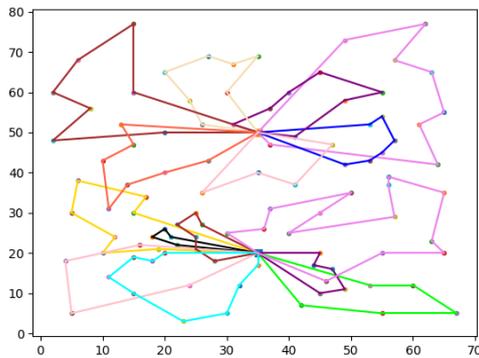
Fonte: Elaborada pelo autor.

A Tabela 13 apresenta a média e o desvio padrão das 30 execuções de cada algoritmo genético. Analisando o desvio padrão, observa-se que as configurações apresentam um desvio padrão pequeno, garantindo que os resultados de qualquer execução dos algoritmos apresentará um resultado próximo da média. Comparando as médias de CD_S20 e

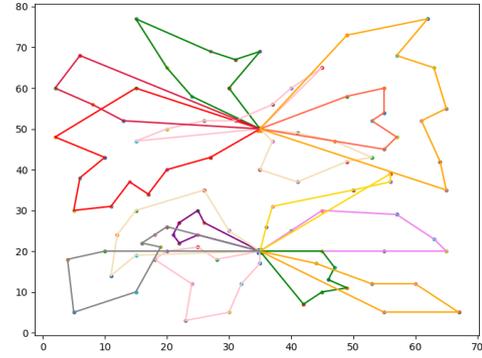
CD_D20, nota-se que em o CD_S20 obteve melhores resultados em 6 das 11 instâncias, mostrando novamente que o algoritmo Clusterização Swap produziu pequenas melhoras em relação à clusterização por distância.

As rotas geradas pelas melhores soluções das três estruturas cliente/depósito para a instância p04 são apresentadas na Figura 29. Os resultados mostram que o algoritmo genético proposto consegue gerar bons resultados, com rotas bem definidas, onde o algoritmo com a Clusterização Swap obteve a melhor solução.

Figura 29: Resultados gerados pelas estruturas para a instância p04



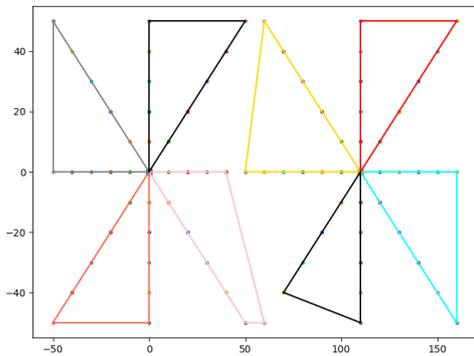
(a) p04 CD_S20 - 1049,20



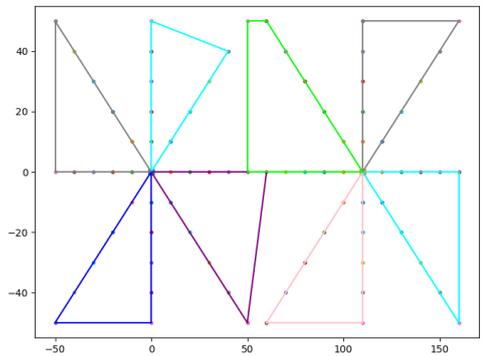
(b) p04 CD_D20 - 1058,00

Fonte: Elaborada pelo autor.

Figura 30: Resultados gerados pelas estruturas para a instância p12



(a) p12 CD_S20 - 1341,84



(b) p12 CD_D20 - 1341,84

Fonte: Elaborada pelo autor.

A Figura 30 apresenta os melhores resultados das rotas geradas para a instância p12. São apresentados o mesmo valor de custo para o CD_S20 e CD_D20. Porém as rotas geradas possuem diferenças mínimas, mostrando que o algoritmo consegue encontrar boas

soluções e que por causa da sua característica aleatória é possível encontrar o mesmo custo com comportamentos diferentes.

O Apêndice A apresenta as imagens das melhores rotas obtidas por CD_S20 e as melhores rotas conhecidas na literatura para as 11 instâncias avaliadas, disponível em (CORDEAU; GENDREAU; LAPORTE, 1997).

5 Conclusão

Neste trabalho foi desenvolvido um Algoritmo Genético para o problema MDVRP, onde os clientes são divididos em estruturas cliente/depósito definidos pelo algoritmo Clusterização Swap e que utiliza um conjunto de operadores existentes na literatura. O algoritmo Clusterização Swap é uma extensão do algoritmo de clusterização por distância que utiliza listas de trocas de clientes entre depósitos para modificar a estrutura cliente/depósito criada pela clusterização por distância. As listas de trocas de clientes entre depósitos proposta neste trabalho é similar as listas propostas em (MAHMUD; HAQUE, 2019; OMBUKI-BERMAN; HANSHAR, 2009), onde a principal diferença é que as listas deste trabalho não permitem que um cliente candidato que esteja distante dos clientes alocados no depósito, pertença a lista deste depósito, garantindo a proximidade dos clientes atendidos pelo depósito. Entre os operadores genéticos implementados, um dos operadores mais importantes é o de mutação entre depósitos que permite modificar a estrutura cliente/depósito, alterando o espaço de busca, utilizando as listas de trocas de clientes entre depósitos. Os algoritmos foram testados utilizando 11 instâncias disponível em (CORDEAU; GENDREAU; LAPORTE, 1997).

Para avaliar o impacto das estruturas cliente/depósito definido na população inicial, foi realizado o Experimento 1. O experimento demonstrou a eficiência do algoritmo genético implementado, mostrando que a estrutura cliente/depósito obtida pela Clusterização Swap consegue trazer uma melhoria comparada a Clusterização de distância.

Como obter uma boa estrutura cliente/depósito para um problema qualquer pode ser difícil, uma forma de ampliar a região de busca do Algoritmo Genético mudando a estrutura cliente/depósito é utilizando a mutação entre depósitos. O Experimento 2 avaliou o impacto da mutação entre depósitos, onde os resultados demonstraram a importância do operador, pois a utilização da mutação permitiu obter melhores resultados.

A mudança proposta nas listas de trocas de clientes entre depósitos foi avaliada no Experimento 3, onde os resultados demonstraram que a remoção de clientes distantes melho-

rou o desempenho do algoritmo ao ser comparado com as listas propostas em (MAHMUD; HAQUE, 2019; OMBUKI-BERMAN; HANSHAR, 2009).

O último experimento avaliou as estruturas cliente/depósitos definidas pelo Algoritmo Swap e pela clusterização por distância utilizando a mutação entre depósitos com as listas de trocas desenvolvidas neste trabalho. Os resultados indicaram a estrutura cliente/depósito definida pelo Algoritmo Swap é capaz de fornecer resultados próximos aos melhores resultados da literatura e trazer resultados melhores que a clusterização por distância para algumas instâncias.

Os resultados obtidos, qualificam o Algoritmo Genético desenvolvido para ser avaliado em problemas reais, o que pode trazer benefícios para o ramo de logística, minimizando os custos operacionais de transporte.

5.1 Trabalhos Futuros

Este trabalho da margem para realização de novas pesquisas portanto, a seguir serão descritos os trabalhos futuros que podem complementar este trabalho:

- Melhorar a clusterização desenvolvida, atualmente é avaliado somente um cliente por vez na lista de trocas. No futuro o algoritmo poderia avaliar vários clientes do mesmo depósito da lista de troca por vez.
- Encontrar alguma métrica que relacione distância, capacidade e tamanho da rota para definir novos critérios de clusterização.
- Verificar instâncias que possuem tamanho máximo de rota como restrição para o MDVRP.
- Aplicar o GA desenvolvido em variantes do MDVRP como o problema de roteamento de veículos com múltiplos depósitos com janela de tempo (MDVRPTW) e o problema de roteamento de veículos com coleta e entrega simultâneas (MDVRPSPD).

Referências

- AROSTEGUI, M. A.; KADIPASAOGLU, S. N.; KHUMAWALA, B. M. An empirical comparison of tabu search, simulated annealing, and genetic algorithms for facilities location problems. *International Journal of Production Economics*, v. 103, n. 2, p. 742–754, 2006.
- AZADEH, A.; FARROKHI-ASL, H. The close–open mixed multi depot vehicle routing problem considering internal and external fleet of vehicles. *Transportation Letters*, v. 11, 02 2017.
- BAI, A.; GAO, H. Solving the multi-depot vehicle routing problem by a genetic algorithm with a new local search procedure. *Expert Systems with Applications*, Elsevier, v. 37, n. 5, p. 3795–3803, 2010.
- BOUANANE, K.; BENADADA, Y.; BENCHEIKH, G. Application of a hybrid genetic algorithm to solve the multi-depots vehicle routing problem with simultaneous delivery and pickup and inventory restrictions. In: *2018 4th International Conference on Logistics Operations Management (GOL)*. [S.l.: s.n.], 2018. p. 1–9.
- BOUYAHYIOUY, K. E.; BELLABDAOUI, A. A new crossover to solve the full truckload vehicle routing problem using genetic algorithm. In: *2016 3rd International Conference on Logistics Operations Management (GOL)*. [S.l.: s.n.], 2016. p. 1–6.
- CHEN, Y. et al. A multi-point crossover genetic algorithm for the vehicle routing problem with time windows. *IEEE Access*, IEEE, v. 5, p. 4381–4392, 2017.
- CORDEAU, J.-F.; GENDREAU, M.; LAPORTE, G. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, v. 30, n. 2, p. 105–119, 1997.
- DARWIN, C. *The Origin of Species: By Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life*. London: John Murray, 1859.
- DAVENDRA, D. *Traveling Salesman Problem, Theory and Applications*. Janeza Trdine 9, 51000 Rijeka, Croatia: InTech, 2010.
- FITRIANA, R.; MOENGIN, P.; KUSUMANINGRUM, U. Improvement route for distribution solutions MDVRP (multi depot vehicle routing problem) using genetic algorithm. *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, v. 528, p. 012042, jun 2019.
- GAO, M.; JIA, Y.; LIU, Z. A novel hybrid genetic algorithm with simulated annealing for the multi-depot vehicle routing problem. *Applied Soft Computing*, Elsevier, v. 39, p. 114–126, 2016.

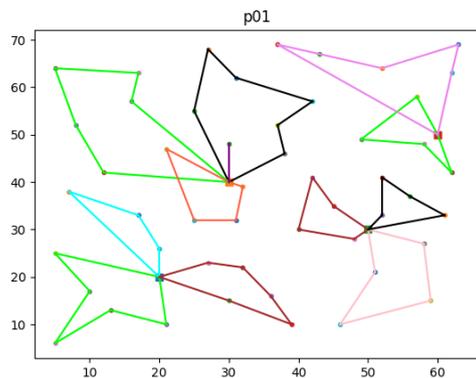
- GAO, X. et al. A hybrid pso-ils algorithm for the multi-depot vehicle routing problem with time windows and capacity constraints. *Information Sciences*, Elsevier, v. 352, p. 19–36, 2016.
- GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st. ed. USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0201157675.
- HAERANI, E. et al. Optimization of multiple depot vehicle routing problem (mdvrp) on perishable product distribution by using genetic algorithm and fuzzy logic controller (flc). In: *2017 5th International Conference on Cyber and IT Service Management (CITSM)*. [S.l.: s.n.], 2017. p. 1–5.
- HE, C.-L.; ZHANG, C.; WANG, X.-Y. A new ant colony algorithm for the multi-depot vehicle routing problem with time windows. *Expert Systems with Applications*, v. 72, p. 205–218, 2017. ISSN 09574174.
- JIANG, J.-x. et al. A hybrid ant colony optimization algorithm for multi-depot vehicle routing problem. *Journal of Supercomputing*, v. 73, n. 2, p. 645–661, 2017. ISSN 1387865X.
- KARAKATIČ, S.; PODGORELEC, V. A survey of genetic algorithms for solving multi depot vehicle routing problem. *Applied Soft Computing*, v. 27, p. 519 – 532, 2015. ISSN 1568-4946.
- KOULAEIAN, M. et al. A multi depot simultaneous pickup and delivery problem with balanced allocation of routes to drivers. *The International Journal of Industrial Engineering: Theory, Applications and Practice*, v. 22, p. 223–242, 01 2015.
- KULKARNI, R.; BHAVE, P. Integer programming formulations of vehicle routing problems. *European Journal of Operational Research*, v. 20, n. 1, p. 58–67, 1985. ISSN 0377-2217.
- LAPORTE, G. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, v. 59, n. 3, p. 345–358, 1992. ISSN 0377-2217.
- LI, B. et al. An adaptive memory-based tabu search algorithm for the multi-depot vehicle routing problem with time windows. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, v. 19, n. 8, p. 2652–2663, 2018.
- LI, J.; LI, Y.; PARDALOS, P. Multi-depot vehicle routing problem with time windows under shared depot resources. *Journal of Combinatorial Optimization*, v. 31, 07 2014.
- LI, Z.; SHI, X.; FAN, Y. Imperialist competitive algorithm with independence and constrained assimilation. *Applied Soft Computing*, Elsevier, v. 78, p. 492–504, 2019.
- LIU, R.; LIANG, X.; YANG, S. A hybrid particle swarm optimization with tabu search for multi-depot vehicle routing problem with time windows. *Expert Systems with Applications*, Elsevier, v. 37, n. 4, p. 3114–3120, 2010.
- LIU, R.; WANG, D. A hybrid genetic algorithm with local search for the multi-depot vehicle routing problem with time windows. *International Journal of Production Economics*, Elsevier, v. 148, p. 96–106, 2014.

- LU, Y.; WU, Y.; ZHOU, Y. Order assignment and routing for online food delivery: Two meta-heuristic methods. In: . [S.l.: s.n.], 2017. p. 125–129.
- MAHMUD, N.; HAQUE, M. M. Solving multiple depot vehicle routing problem (mdvrp) using genetic algorithm. In: IEEE. *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. [S.l.], 2019. p. 1–6.
- MESA, J. A.; BIEDERMANN, D. P. A variable neighborhood search heuristic for the multi-depot vehicle routing problem. *Computers Operations Research*, Elsevier, v. 33, n. 1, p. 274–292, 2006.
- MIDAOU, M. el; QBADOU, M.; MANSOURI, K. Logistics chain optimization and scheduling of hospital pharmacy drugs using genetic algorithms: Morocco case. *International Journal of Web-Based Learning and Teaching Technologies*, v. 16, p. 54, 03 2021.
- MOHAMMED, M. A. et al. Solving vehicle routing problem by using improved genetic algorithm for optimal solution. *Journal of Computational Science*, v. 21, p. 255 – 262, 2017. ISSN 1877-7503.
- MOKHTAR, E.; ELHOSENY, M.; HASSANIEN, A. E. A hybrid heuristic for the multi-depot vehicle routing problem. *Journal of Intelligent Fuzzy Systems*, IOS Press, v. 27, n. 2, p. 859–870, 2014.
- NARASIMHA, K. S. V.; KIVELEVITCH, E.; KUMAR, M. Ant colony optimization technique to solve the min-max multi depot vehicle routing problem. In: *2012 American Control Conference (ACC)*. [S.l.: s.n.], 2012. p. 3980–3985.
- OMBUKI-BERMAN, B.; HANSHAR, F. T. Using genetic algorithms for multi-depot vehicle routing. In: *Bio-inspired algorithms for the vehicle routing problem*. [S.l.]: Springer, 2009. p. 77–99.
- OPOKU-AMANKWAAH, A.; OMBUKI-BERMAN, B. An age layered population structure genetic algorithm for the multi-depot vehicle routing problem. In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. [S.l.: s.n.], 2017. p. 1–8.
- OUDOUAR, F.; FALLAHI, A. E.; ZAOUI, E. An improved heuristic based on clustering and genetic algorithm for solving the multi-depot vehicle routing problem. *International Journal of Recent Technology and Engineering*, v. 8, n. 3, p. 6535–6540, 2019. Cited By 2.
- RABBANI, M. et al. A hybrid genetic algorithm for multi-depot vehicle routing problem with considering time window repair and pick-up. *Journal of Modelling in Management*, v. 13, p. 698–717, 09 2018.
- RENAUD, J.; LAPORTE, G.; BOCTOR, F. F. A tabu search heuristic for the multi-depot vehicle routing problem. *Computers Operations Research*, v. 23, n. 3, p. 229–235, 1996. ISSN 0305-0548.
- S, S.; JOSEPH, K. S.; JAYAKUMAR, S. A hybrid population seeding technique based genetic algorithm for stochastic multiple depot vehicle routing problem. In: *2015 International Conference on Computing and Communications Technologies (ICCCT)*. [S.l.: s.n.], 2015. p. 119–127.

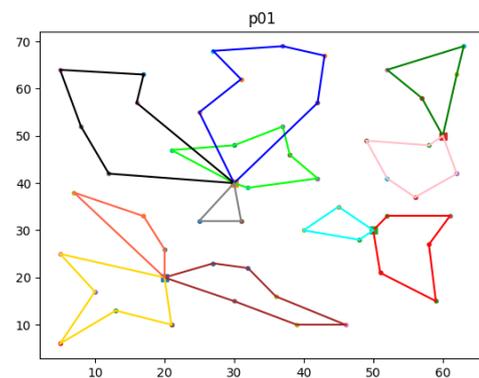
- SHANG, W. et al. A tabu search algorithm for the multi-depot vehicle routing problem with time windows and multiple products. *IEEE Transactions on Automation Science and Engineering*, IEEE, v. 12, n. 1, p. 93–105, 2015.
- SINGH, V.; L., G.; PUNDIR, A. An improved genetic algorithm for solving multi depot vehicle routing problems. *International Journal of Information Systems and Supply Chain Management*, v. 12, p. 1–26, 10 2019.
- SONG, G.; WEN, G.; ZHANG, G. A hybrid genetic algorithm for the multi-depot vehicle routing problem with time windows. *Applied Soft Computing*, Elsevier, v. 71, p. 613–625, 2018.
- TALBI, E.-G. *Metaheuristics: from design to implementation*. [S.l.]: John Wiley & Sons, 2009.
- TAUFIK, T.; RAMADHAN, B.; NABABAN, E. A hybrid genetic algorithm with local search for the multi-depot vehicle routing problem with stochastic demands. *Procedia Computer Science*, Elsevier, v. 59, p. 207–215, 2015.
- WANG, K.; XU, D.-l.; XU, Y.-y. A Multi-Depot Vehicle Routing Problem with Time Windows: An Improved Ant Colony Optimization Approach. *Multimedia Tools and Applications*, v. 76, n. 14, p. 7119–7136, 2016. ISSN 1380-7501.
- WANG, S. et al. Fitness-scaling adaptive genetic algorithm with local search for solving the multiple depot vehicle routing problem. *SIMULATION*, v. 92, n. 7, p. 601–616, 2016.
- WANG, W. et al. A multi-objective hybrid genetic algorithm for the multi-depot vehicle routing problem with time windows. *IEEE Access*, IEEE, v. 7, p. 23611–23626, 2019.
- WANG, Y. et al. Collaborative multi-depot logistics network design with time window assignment. *Expert Systems with Applications*, v. 140, p. 112910, 2020. ISSN 0957-4174.
- WEN, M. et al. A Multi-Depot Vehicle Routing Problem with Time Windows: An Ant Colony Optimization Approach. *Journal of Intelligent Manufacturing*, v. 22, n. 6, p. 1297–1307, 2011. ISSN 0925-5273.
- WU, J. et al. Solving multi-depot vehicle routing problem with time windows using a genetic algorithm with elitist preserving and diversity enhancing strategies. *IEEE Access*, IEEE, v. 6, p. 31028–31037, 2018.
- ZHANG, D. et al. An ant colony optimization technique for solving min-max multi-depot vehicle routing problem. *Mathematical Problems in Engineering*, Hindawi, v. 2019, p. 5878960, 2019.
- ZHANG, X.; HUANG, H.; ZHANG, Q. A modified particle swarm optimization algorithm for multi-depot vehicle routing problem with time windows and capacity constraints. *Expert Systems with Applications*, Elsevier, v. 38, n. 3, p. 1689–1697, 2011.
- ZHANG, Z. et al. A multi-start local search based hybrid genetic algorithm for multi-depot vehicle routing problem. *IEEE Access*, IEEE, v. 7, p. 162194–162205, 2019.

APÊNDICE A

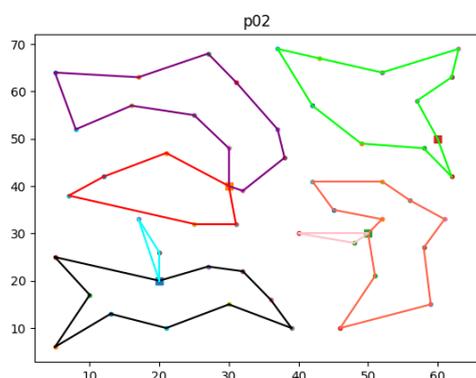
Este apêndice contém a representação de rotas para o problema de roteamento de veículos com múltiplos depósitos de 11 instâncias disponíveis na literatura (CORDEAU; GENDREAU; LAPORTE, 1997). São apresentadas as rotas alcançadas pelo algoritmo genético desenvolvido nesse trabalho denominado CD_S20 e a melhor solução encontrada na literatura disponível em (CORDEAU; GENDREAU; LAPORTE, 1997).



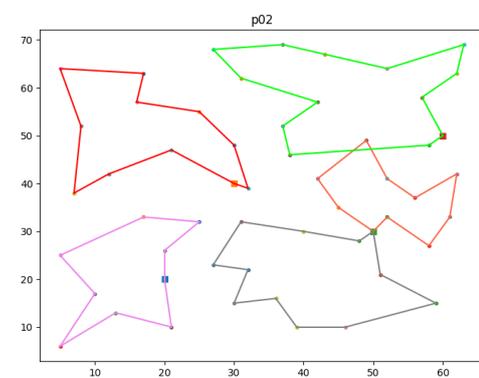
(c) p01 CD_S20 - 606,68



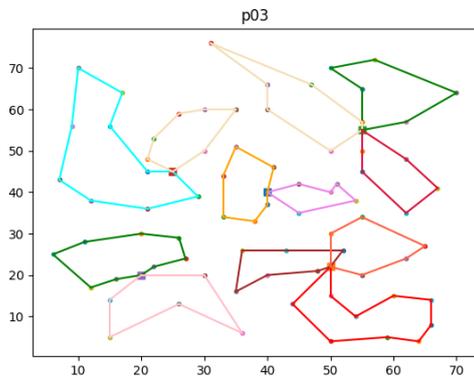
(d) p01 Literatura - 576,87



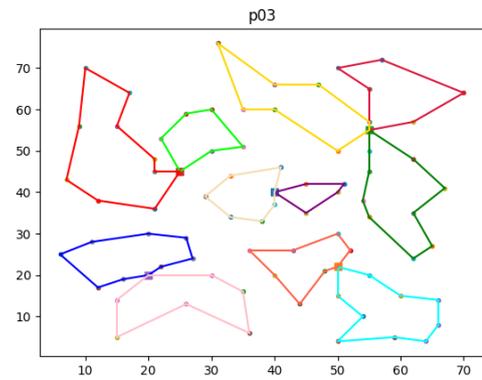
(e) p02 CD_S20 - 482,36



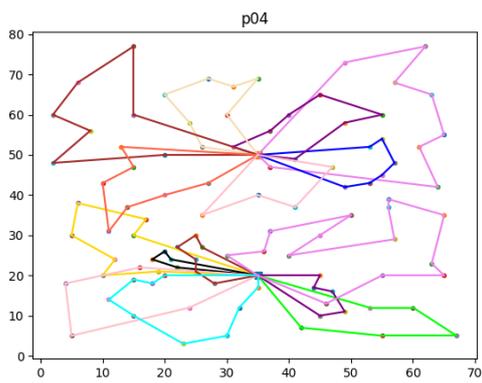
(f) p02 Literatura - 473,53



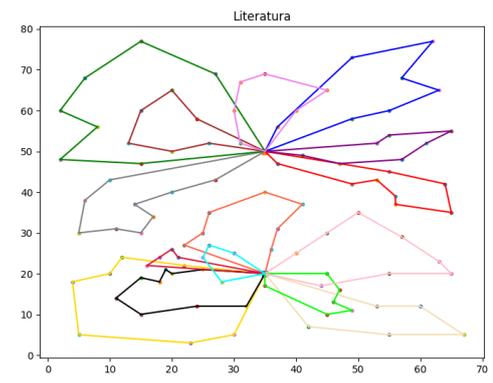
(g) p03 CD_S20 - 663,56



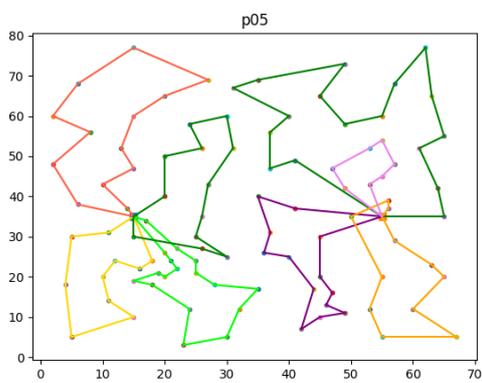
(h) p03 Literatura - 641,19



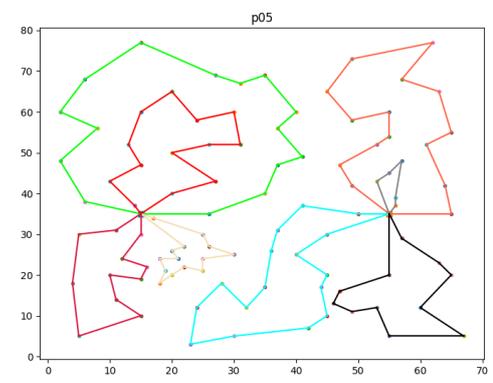
(i) p04 CD_S20 - 1049,20



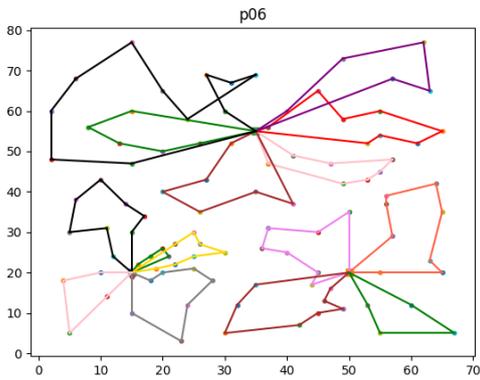
(j) p04 Literatura - 1001,59



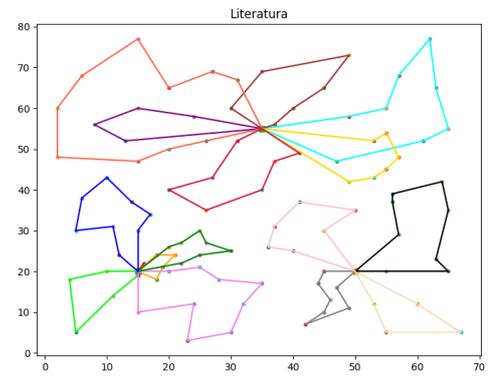
(k) p05 CD_S20 - 793,46



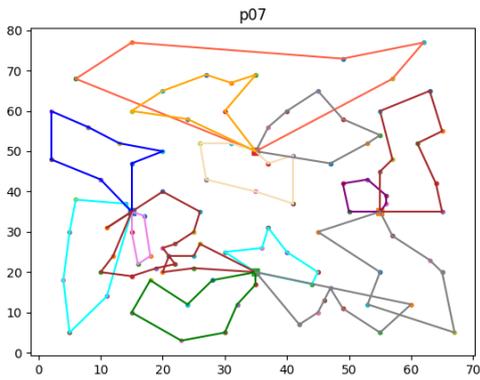
(l) p05 Literatura - 750,03



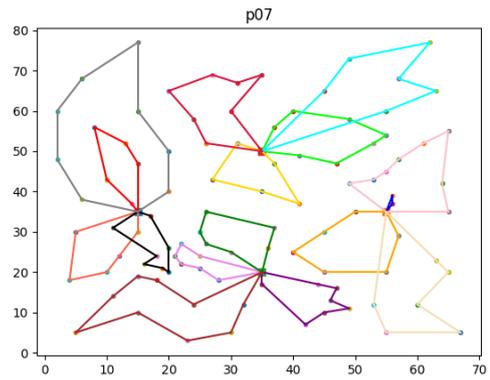
(m) p06 CD_S20 - 917,18



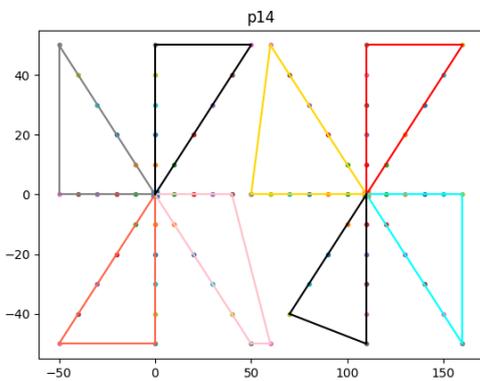
(n) p06 Literatura - 876,50



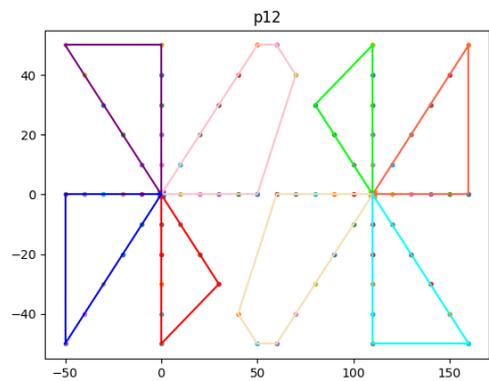
(o) p07 CD_S20 - 927,15



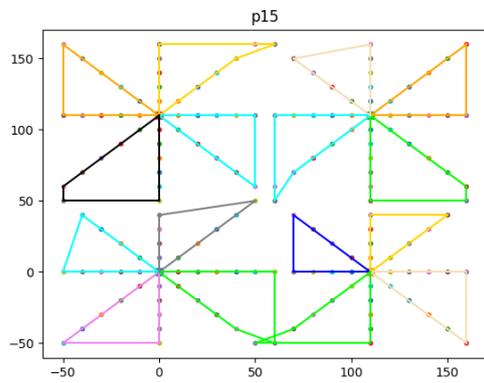
(p) p07 Literatura - 885,80



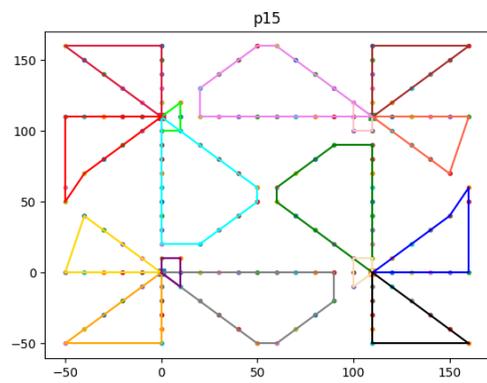
(q) p12 CD_S20 - 1341,84



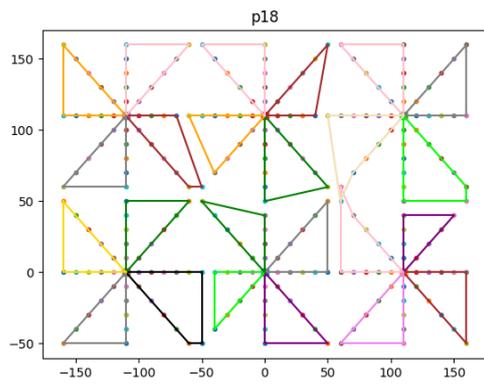
(r) p12 Literatura - 1318,95



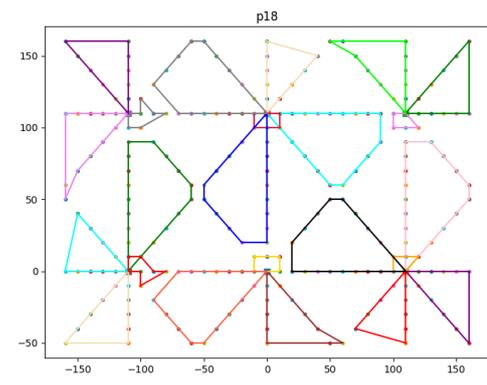
(s) p15 CD_S20 - 2681,78



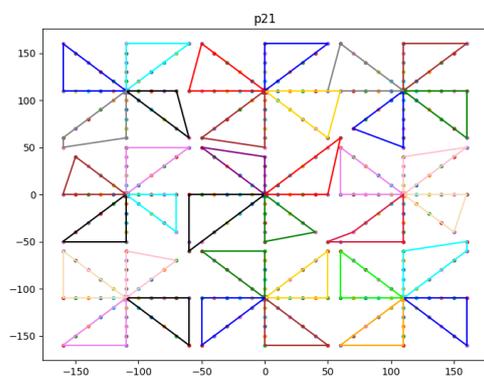
(t) p15 Literatura - 2505,42



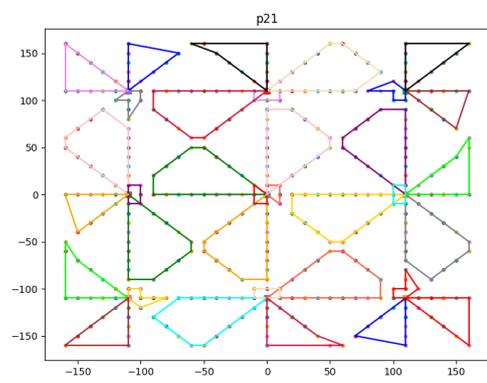
(u) p18 CD_S20 - 4055,56



(v) p18 Literatura - 3702,85



(w) p21 CD_S20 - 6118,34



(x) p21 Literatura - 5474,84