

UNIVERSIDADE DO VALE DO RIO DOS SINOS — UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA
NÍVEL MESTRADO

ROBERTO DA SILVA ARAUJO

UMA ABORDAGEM HÍBRIDA PARA A SOLUÇÃO DO PROBLEMA DE ROTEAMENTO
DE VEÍCULOS COM MÚLTIPLOS DEPÓSITOS E FROTA HETEROGÊNEA
Algoritmo Genético e Busca Tabu

SÃO LEOPOLDO
2017

Roberto da Silva Araujo

UMA ABORDAGEM HÍBRIDA PARA A SOLUÇÃO DO PROBLEMA DE ROTEAMENTO
DE VEÍCULOS COM MÚLTIPLOS DEPÓSITOS E FROTA HETEROGÊNEA
Algoritmo Genético e Busca Tabu

Dissertação apresentada como requisito parcial
para a obtenção do título de Mestre pelo
Programa de Pós-Graduação em Computação
Aplicada da Universidade do Vale do Rio dos
Sinos — UNISINOS

Orientador:
Prof. Dr. Arthur Tórgo Gómez

São Leopoldo
2017

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)

A663a Araujo, Roberto da Silva

UMA ABORDAGEM HÍBRIDA PARA A SOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM MÚLTIPLOS DEPÓSITOS E FROTA HETEROGÊNEA: algoritmo genético e busca tabu / Roberto da Silva Araujo — 2017.

84 f.: il.; 30 cm.

Dissertação (Mestrado em Computação Aplicada) — Programa de Pós-Graduação em Computação Aplicada, Universidade do Vale do Rio dos Sinos UNISINOS.

Orientador: Prof. Dr. Arthur Tórgo Gómez.

1. Informática. 2. Algoritmo Genético. 3. Busca Tabu.
4. Problema de Roteamento de Veículo. 5. Múltiplos Depósitos.
6. Frota Heterogênea. I. Gómez, Arthur Tórgo II. Título.

CDD 005.1

Bibliotecária responsável: Elisete Sales de Souza — CRB 10/1441

(Esta folha serve somente para guardar o lugar da verdadeira folha de aprovação, que é obtida após a defesa do trabalho. Este item é obrigatório, exceto no caso de TCCs.)

Ao meu orientador, Dr. Arthur Tórgo Gómez, pelos ensinamentos e orientações na vida acadêmica.

À minha filha Paula Franz, pela compreensão nos momentos difíceis.

À minha mãe Maria Antônia da Silva Araujo, pela criação e educação.

Aos meus irmãos e sobrinhos pelos pensamentos positivos.

Ao meu pai Nilson Jobim Araujo em memória.

Aos meus colegas de trabalho, pelo apoio e incentivo.

À Deus.

RESUMO

Neste trabalho é apresentado um Algoritmo Híbrido (AH) aplicado ao Problema de Roteamento de Veículos com Múltiplos Depósitos e Frota Heterogênea (PRVMDFH). Um conjunto de clientes é atendido por um número fixo de veículos de diferentes tamanhos, por múltiplos depósitos, sujeito a restrições da capacidade do veículo. As meta-heurísticas utilizadas na construção do AH são o Algoritmo Genético (AG) e a Busca Tabu (BT). O AG usa operadores de cruzamento Mapeado Parcialmente (PMX), Cromossomo de Duas Partes (TCX) e de Ordem (OX), o operador de mutação Troca e o Algoritmo de Busca Local (ABL). A BT usa os métodos de Troca, Retirada e Inserção e Deslocamento. São analisadas as políticas de diversificação no ABL e a intensificação com as estratégias de geração de vizinhança na BT. São utilizados conjuntos de dados padrões de problemas testes, para executar os algoritmos propostos. Os resultados obtidos, comparados a outros autores, apresentaram boas soluções para diferentes tamanhos de problemas testes.

Palavras-chave: Informática. Algoritmo Genético. Busca Tabu. Problema de Roteamento de Veículo. Múltiplos Depósitos. Frota Heterogênea.

ABSTRACT

This paper presents a Hybrid Algorithm (AH) applied to the Problem of Vehicle Routing with Multiple-Deposit and Heterogeneous Fleet (MDFHPRV). A number of customers are served with a fixed number of vehicles of different sizes by multi-depot, subject to capacity constraints of the vehicle. The meta-heuristics used in AH construction are Genetic Algorithm (GA) and Tabu Search (BT). The GA uses Partial Mapped Crossover (PMX), Two Part Chromosome Crossover (TCX) and Order Crossover (OX) operators, the Exchange mutation operator, and the Local Search Algorithm (ABL). BT uses the Exchange, Retrieve, and Insertion and Displacement methods. Diversification policies were analyzed in the ABL and the intensification with the neighborhood generation strategies in BT. Standard sets of test problems are used to execute the proposed algorithms. The obtained results, compared to other authors, presented good solutions for different sizes of test problems.

Keywords: Computer. Genetic Algorithm. Tabu Search. Vehicle Routing Problem. Multi-depot. Heterogeneous Fleet.

LISTA DE FIGURAS

Figura 1:	Representação do cromossomo	26
Figura 2:	Representação do cromossomo com a sequencia da rota	26
Figura 3:	Representação do cromossomo sem a sequênci da rota	27
Figura 4:	Representação do cromossomo	27
Figura 5:	Operador OX	28
Figura 6:	Operador de Cruzamento PMX	29
Figura 7:	Operador de Cruzamento TCX	30
Figura 8:	Operador de Mutação Troca	31
Figura 9:	Operador Remove e Insere	32
Figura 10:	Último será o primeiro LWBF	32
Figura 11:	Inversão	32
Figura 12:	Depósitos, Veículos e Clientes	41
Figura 13:	Arquivo do Problema Teste	45
Figura 14:	Montagem da rede	46
Figura 15:	Representação da Rede	46
Figura 16:	Solução Inicial	47
Figura 17:	Arquitetura do AH	48
Figura 18:	Arquitetura do AG	50
Figura 19:	Representação do cromossomo	51
Figura 20:	Estrutura para o Cálculo da Aptidão	51
Figura 21:	Exemplo de uma estrutura que armazena a roleta	53
Figura 22:	PMX. Filho 1, Veículos e Depósitos	54
Figura 23:	TCX. Filho com Clientes, Veículos e Depósitos	54
Figura 24:	OX. Filho com Clientes, Veículos e Depósitos	55
Figura 25:	Mutação Troca	55
Figura 26:	Fluxograma demonstrando a sequênci do ABL.	56
Figura 27:	Movimento Remove Insere Depois	57
Figura 28:	Código Fonte do ABL	58
Figura 29:	Arquitetura da execução separada da BT	60
Figura 30:	Vizinhança	61
Figura 31:	Lista de candidatos para o Movimento Troca	62
Figura 32:	Lista de candidatos para o Movimento Retira e Insere	63
Figura 33:	Lista de candidatos para o Movimento Deslocamento	63
Figura 34:	Inclusão na LT	64
Figura 35:	Solução origem	65
Figura 36:	Solução destino	65
Figura 37:	Variações do parâmetro iterações no Algoritmo de Busca Local	71
Figura 38:	Variações do Tamanho da LT (tamLista)	72
Figura 39:	Variações do número de interações sem melhora (nbMax)	73

LISTA DE TABELAS

Tabela 1:	Dados do AG comparado a Cordeau, Gendreau e Laporte (1997)	70
Tabela 2:	Variações do parâmetro Número de Gerações	70
Tabela 3:	Variações do parâmetro iterações no Algoritmo de Busca Local	71
Tabela 4:	Variações do movimento de vizinhança Busca Tabu	72
Tabela 5:	Variações do Tamanho da LT	72
Tabela 6:	Variações do número de interações sem melhora (nbMax)	73
Tabela 7:	Dados Algoritmo Híbrido comparado a Desrochers, Desrosiers e Solomon (1992)	74
Tabela 8:	Dados do AH comparado a Cordeau, Gendreau e Laporte (1997)	75
Tabela 9:	Médias obtidas na execução do AH sem Custo Fixo do Veículo	75
Tabela 10:	Comparação entre AG, BT e AH para o problema teste p04	76
Tabela 11:	Comportamento da Capacidade dos Tipos de Veículos com o AH	77
Tabela 12:	Valores para o Custo Fixo dos Veículos com AH	78
Tabela 13:	Resultado obtidos para Cordeau, Gendreau e Laporte (1997) com Custo Fixo na Função Objetivo	78

LISTA DE SIGLAS

AG	Algoritmo Genético
AH	Algoritmo Híbrido
MH	Modelo Híbrido
BT	Busca Tabu
ABL	Algoritmo de Busca Local
CA	Critério de Aspiração
CP	Critério de Parada
DNA	Ácido Desoxirribonucleico
FO	Função Objetivo
GRASP	Procedimento de Busca Adaptativa Gulosa e Randomizada
HC	Heurística Construtiva
HM	Heurística de Melhoramento
LT	Lista Tabu
LC	Lista de Candidatos
LWBF	Último será o Primeiro
MA	Memória Adaptativa
MPRVDH	Modelo do Problema de Roteamento de Veículo com Múltiplos Depósitos e Frota Heterogênea
MTB	Método torneio binário
OX	Cruzamento de Ordem
PRVC	Problema de Roteamento de Veículos Capacitado
PCV	Problema do Caixeiro Viajante
PRV	Problema de Roteamento de Veículos
PRVA	Problema de Roteamento de Veículos Aberto
PRVMD	Problema de Roteamento de Veículos com Múltiplos Depósitos
PRVSCE	Problema de Roteamento De Veículos com Coleta e Entrega Simultâneas
PRVFH	Problema de Roteamento de Veículos com Frota Heterogênea
PRVMVA	Problema de Roteamento de Veículos com Múltipla Viagem e Agendamento
PRVDE	Problema de Roteamento de Veículos com Demanda Estocástica
PRVJT	Problema de Roteamento de Veículos com Janela de Tempo
PR	Path Relinking
PMX	Cruzamento Mapeado Parcialmente
RAR	Remove e Re-Insere

RR	Roda de Roleta
SA	Simulated Annealing
SS	Scatter Search
TT	Tabu Tenure
TCX	Cruzamento do Cromossomo de Duas Partes
VNS	Variable Neighborhood Search

SUMÁRIO

1	INTRODUÇÃO	14
2	O PROBLEMA DE ROTEAMENTO DE VEÍCULOS (PRV)	16
2.1	PRV Clássico	17
2.2	Principais Variações do PRV	19
2.2.1	PRV Capacitado	19
2.2.2	PRV com Janelas de Tempo	19
2.2.3	PRV Aberto	20
2.2.4	PRV Múltiplos Depósitos	20
2.2.5	PRV com Coleta e Entrega Simultânea	20
2.2.6	PRV com Frota Heterogênea	21
2.2.7	PRV com Múltipla Viagem e Problema de Agendamento	21
2.2.8	PRV com Demanda Estocástica	21
3	META-HEURÍSTICAS	22
3.1	Introdução à Meta-Heurística	22
3.2	Heurística Híbrida	22
3.3	Algoritmo Genético (AG)	23
3.3.1	Introdução ao Algoritmo Genético	23
3.3.2	Geração da População Inicial	23
3.3.3	Método de Seleção	25
3.3.4	Codificação	25
3.3.5	Função de Aptidão	27
3.3.6	Cruzamento e Recombinação	28
3.3.7	Mutação	30
3.3.8	Troca	31
3.4	Busca Tabu (BT)	32
3.4.1	Introdução à Busca Tabu	33
3.4.2	Solução Inicial	34
3.4.3	Lista Tabu	34
3.4.4	Tabu Tenure	35
3.4.5	Critério de Aspiração	36
3.4.6	Critério de Parada	36
3.4.7	Lista de Candidatos	36
3.4.8	Estrutura de Vizinhança	37
3.4.9	Diversificação e Intensificação na BT	38
4	MODELO	40
4.1	Premissas	40
4.2	Formulação Matemática	42
4.3	Arquitetura do Modelo Híbrido (MPRVDH)	44
4.3.1	Montar a Rede	44
4.3.2	Geração da Solução Inicial	46
4.3.3	Algoritmo Híbrido AH	47
4.3.4	Execução somente do AG	49
4.3.5	Execução somente do BT	60

5 PLANEJAMENTO DOS EXPERIMENTOS	66
5.1 Ajuste de Parâmetros	66
6 RESULTADOS	68
6.1 Conjunto de Dados	68
6.2 Resultados Obtidos	69
6.2.1 AG com Múltiplos Depósitos e Um Tipo de Veículo	69
6.2.2 BT com Múltiplos Depósitos e Um Tipo de Veículo	71
6.2.3 AH Depósitos e Um ou Três Tipos de Veículos	73
7 CONSIDERAÇÕES FINAIS	79
7.1 Conclusão	79
7.2 Trabalhos Futuros	80
REFERÊNCIAS	81

1 INTRODUÇÃO

O Problema de Roteamento de Veículos (PRV) é um problema de Otimização Combinatória e tem sido amplamente estudado por diferentes pesquisadores, durante as últimas décadas KUMAR; PANNEERSELVAM, 2012; CORDEAU et al., 2006. Boas soluções foram geradas para resolver este problema, tanto na versão clássica do PRV e também nas diferentes variações e aplicações já estudadas para ele (ANBUUDAYASANKAR et al., 2012; BRANDÃO, 2009; MONTANÉ; GALVAO, 2006; SIMAS; GÓMEZ, 2007; BELHAIZA; HANSEN; LAPORTE, 2014; FLESZAR; OSMAN; HINDI, 2009).

Atualmente, muitos estudos vêm sendo realizados, na construção de algoritmos, para a resolução de problemas na área de Otimização Combinatória. Existem alguns problemas que necessitam de métodos, que possibilitem um resultado viável computacionalmente. Entre eles, estão o PRV, o Problema do Caixeiro Viajante, o Problema da Mochila e o Problema do Corte Guilhotinado, por exemplo. Estes problemas são de difícil resolução computacional, já que é necessária a avaliação de um conjunto de rotas possíveis.

Para a resolução dos problemas de otimização combinatória de larga escala, os métodos exatos não são eficientes em tempo computacional aceitável (SIMAS; GÓMEZ, 2007), (PAPADIMITRIOU; STEIGLITZ, 1982), (GLOVER; LAGUNA, 2013). As heurísticas são métodos mais apropriados para a resolução dos problemas de otimização combinatória, pois, apesar de não garantirem o resultado ótimo, se tornam mais eficientes, pois não percorrem todo o espaço de busca de soluções, reduzindo assim, o tempo de execução computacional.

As empresas de logística buscam melhorar sua competitividade, cada vez mais, a fim de diminuir os custos, por exemplo, com o transporte que, é responsável por entregar e receber a matéria prima e os produtos. A fim de distribuírem os produtos, as empresas armazenam estes, em estruturas denominadas depósitos. Estas necessitam fazer o planejamento do roteiro de entregas dos produtos, que estão armazenados nos depósitos, com a utilização dos veículos, para atender os pedidos de seus clientes. Este planejamento é feito no sentido de atender os clientes da forma mais econômica possível, levando em consideração a distância percorrida, o tempo de viagem e serviço de entrega e os custos fixos e variáveis do veículo. Dentre outros, ocorrem as restrições que podem ser a capacidade do veículo, a janela de tempo do cliente para entrega e o tipo de frota (FLESZAR; OSMAN; HINDI, 2009). As empresas necessitam saber de forma eficiente e rápida, qual a maneira mais econômica de montar os roteiros de entregas, que são denominadas rotas. A construção das rotas, para atender mais de 50 clientes, caracteriza-se como um problema NP-completo (PAPADIMITRIOU; STEIGLITZ, 1982). Segundo Papadimitriou e Steiglitz (1982), o significado prático da notação de um problema NP-completo define que problemas são em sua essência intratáveis do ponto de vista computacional, já que não são facilmente tratados com soluções de algoritmos eficientes. Com isso, qualquer algoritmo que resolve corretamente um problema NP-completo, poderá apresentar-se de forma impraticável, para problemas de larga escala. Difundidas, a partir da década de 1990, surgem as Meta-heurísticas que

se referem a estratégias que orientam e modificam outras heurísticas, na busca de resolução de problemas de maior complexidade computacional (GLOVER; LAGUNA, 2013). Este cenário caracteriza o PRV abordado nos trabalhos de Anbuudayasankar et al. (2012); Brandão (2009); Montané e Galvao (2006); Simas e Gómez (2007); Belhaiza, Hansen e Laporte (2014); Fleszar, Osman e Hindi (2009). A maior parte dos algoritmos utilizados na resolução do PRV são os algoritmos de aproximação (BRANDÃO, 2004).

O objetivo deste trabalho é desenvolver um modelo computacional, através de um Algoritmo Híbrido (AH) utilizando as meta-heurísticas BT e AG. Este modelo considera o PRV com Múltiplos Depósitos e Frota Heterogênea, observando as restrições da capacidade do veículo, do retorno ao depósito, de apenas uma visita ao cliente, do atendimento do cliente por um depósito, do limite de depósitos, e do limite dos tipos de veículos. Quanto aos objetivos específicos pretende-se: (i) implementar um AG para PRV, (ii) implementar operadores de mutação e cruzamento para o algoritmo AG, (iii) implementar um algoritmo de busca local no AG, (iv) implementar um algoritmo BT, (v) implementar heurísticas para geração de vizinhança, (vi) implementar o AH composto pelo AG e BT e (vii) desenvolver estratégias para Intensificação e Diversificação no AG e BT.

A contribuição deste trabalho é abordar o PRV levando em consideração mais de um depósito e frotas de veículos heterogêneos, de forma simultânea. Considerou-se as restrições como capacidade do veículo, retorno de veículo ao depósito, visita ao cliente apenas uma vez, atendimento do um cliente por um só depósito, o limite do número de depósitos e o limite dos tipos de veículos. Foi desenvolvido no AG um algoritmo de busca local. Na BT utilizou-se diferentes estratégias para geração de vizinhanças. Foram implementados os operadores para cruzamento PMX, TCX e OX e os operadores de mutação, no AG.

O trabalho está organizado da seguinte forma: no capítulo 2 é contextualizado o PRV e seus conceitos, as diferentes variações já estudadas e a formulação matemática. O capítulo 3 apresenta as metas-heurísticas utilizadas no PRV e uma visão sobre algoritmos híbridos, uma visão sobre Algoritmo Genético e sobre a meta-heurística BT, já no capítulo 4 a descrição do MPRVDH Modelo do Problema de Roteamento de Veículo com Múltiplos Depósitos e Frota Heterogênea. É abordado, no capítulo 5, o planejamento dos experimentos, já no capítulo 6 são mostrados os conjuntos de dados de referência e os resultados obtidos e, no capítulo 7, as considerações finais.

2 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS (PRV)

O PRV é frequentemente descrito como o problema em que os veículos tendo como base um depósito central, são obrigados a visitar clientes geograficamente dispersos, a fim de satisfazer seus pedidos. O problema é construir um conjunto viável de rotas, uma para cada veículo, de maneira que a distância total percorrida por cada veículo seja a menor possível. A rota é uma sequência de locais que um veículo deve visitar, juntamente com a indicação do serviço que presta (MIRABI, 2015). O custo está relacionado a um objetivo, que pode ser a distância ou o tempo, por exemplo. Há uma grande variedade de situações e, portanto, o problema não é único, mas uma vasta classe de problemas, cada um com suas próprias características e limitações (BRANDAO; MERCER, 1997) e (GALAFASSI, 2011).

Conforme Simas e Gómez (2007), o PRV pode ser descrito como um conjunto de clientes que deve ser atendido por uma frota de veículos, satisfazendo algumas restrições. No PRV, existem aplicações muito importantes na área de distribuição e transporte, com a entrega de mercadorias e informações entre depósitos e clientes.

Conforme Cordeau et al. (2002), nenhum algoritmo exato é capaz de resolver consistentemente casos com mais de 50 clientes. O PRV é um problema de otimização combinatória difícil, e apenas problemas de pequena escala, podem ser resolvidos com algoritmos exatos. Em geral, as rotas obedecem um período de tempo definido pelo cliente, que especifica o horário de início e fim em que as entregas devem ser feitas. Às vezes não é realista que um pedido de um cliente deva ser entregue por um único veículo. Ao permitir que as entregas sejam divididas, um cliente pode ser servido por mais de um veículo. O objetivo consiste em minimizar o número de veículos e a distância total percorrida (HO; HAUGLAND, 2004).

Diversos autores já abordaram o Problema de Roteamento de Veículos (PRV) em suas diferentes variações. O PRV clássico tem trabalhos publicados conforme Cordeau et al. (2006) e Brandão (2004), Glover e Kochenberger (2006), Simas e Gómez (2007), Pisinger e Ropke (2007), Fleszar, Osman e Hindi (2009) e Belhaiza, Hansen e Laporte (2014), entre outros. Especificamente, sobre o Problema de Roteamento de Veículos capacitado, os seguintes autores escreveram sobre este tema: Augerat et al. (1998), Cordeau et al. (2002) e Escobar et al. (2014). Tratando a respeito de PRV com coleta e entrega um dos trabalhos publicados foi escrito por Montané e Galvao (2006).

Os autores Subramanian et al. (2012), Herrero et al. (2014) consideram em seu trabalho o PRV com Frota Heterogênea. Tratando a abordagem de múltiplos depósitos, os seguintes autores publicaram artigos a respeito: Escobar et al. (2014) e Mirabi (2015). Os autores Dayarian et al. (2015) publicaram sobre o PRV múltiplos períodos, Henke, Speranza e Wäscher (2015) publicaram sobre múltiplos compartimentos e Tan, Cheong e Goh (2007) sobre múltiplos objetivos. Brandao e Mercer (1997), em seu trabalho, trataram sobre múltipla viagem, que consiste em um veículo atender um cliente com mais de uma viagem num mesmo dia de trabalho. O PRV pode ser tratado também como simétrico ou assimétrico. Considera-se que os clientes for-

mam uma rede, onde os nodos são as localizações, e que as ligações entre nodos são caminhos de ida e volta. Na versão simétrica do problema, a distância entre cada par de localizações é a mesma nas duas direções, ao passo que na versão assimétrica, não é possível afirmar que entre cada par de localizações, a distância seja a mesma (TOTH; VIGO, 1999). Uma abordagem simétrica e assimétrica backhauls foi tratada por Toth e Vigo (1999).

2.1 PRV Clássico

O PRV clássico consiste em um depósito central, sendo que, os pedidos dos clientes são atendidos por uma frota de veículos com capacidade limitada. As rotas começam e terminam no depósito. Um cliente possui um tempo de serviço para receber a entrega. O objetivo é minimizar a distância total percorrida, considerando que, cada cliente seja atendido exatamente uma vez por um único veículo associado a um percurso, a carga total dos pedidos dos clientes não pode exceder a capacidade do veículo e por fim, a duração do tempo de viagem mais o tempo de serviço, respeitem um valor pré-estabelecido (BERGER; BARKAOUI, 2003) (LI; GOLDEN; WASIL, 2007).

O PRV Clássico pode ser formalmente definido como se segue. Seja $G(V, A)$ um grafo direcionado ou não, onde $V = \{v_0, v_1, \dots, v_n\}$ é um conjunto de vértices, e $A = (v_i, v_j) : v_i, v_j \in V, i \neq j \text{ e } i < j$. O vértice v_0 representa um depósito, enquanto que os vértices restantes correspondem aos clientes. Com A estão associados os nodos à matriz de custo (C_{ij}) e uma matriz de tempo de viagem (T_{ij}) . Se essas matrizes são simétricas, como é comumente o caso, então o padrão para definir o PRV é um grafo não direcionado $G(V, E)$, onde $E = \{(v_i, v_j) : v_i, v_j \in V, i < j\}$ é um conjunto de arestas (KUMAR; PANNEERSELVAM, 2012). Segue a formulação do PRV clássico.

As principais variáveis de decisão são:

T_i tempo de chegada no nodo i ; w_i tempo de espera no nodo i ;

$x_{ijk} \in \{0, 1\}$ onde, $x_{ijk} = 0$ se não houver arco do nó i para o nó j e $x_{ijk} = 1$ caso contrário.

$i \neq j; i < j; i, j \in \{0, 1, 2, \dots, N\}$

Parâmetros utilizados:

K o número total de veículos;

N o número total de clientes;

c_{ij} o custo atribuído ao arco do nodo i para o nodo j ;

t_{ij} o tempo de viagem entre o nodo i e o nodo j ;

m_i a demanda do nodo i ;

q_k a capacidade do veículo k ;

e_i o tempo inicial de chegada no nodo i ;

l_i o tempo final de partida do nodo i ;

f_i o tempo de serviço no nodo i ;

r_k o tempo máximo de rota permitido para o veículo k .

$$\text{Minimize } F = \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^K C_{ij} x_{ijk} \quad (2.1)$$

Sujeito à:

$$\sum_{k=1}^N \sum_{j=1}^N x_{ijk} \leq K, \text{ para } i = 0; \quad (2.2)$$

$$\sum_{j=1}^N x_{ijk} = 1, \text{ para } i = 0 \text{ e } k \in \{1, \dots, K\}; \quad (2.3)$$

$$\sum_{j=1}^N x_{jik} = 1, \text{ para } i = 0 \text{ e } k \in \{1, \dots, K\}; \quad (2.4)$$

$$\sum_{k=1}^K \sum_{j=0, j \neq i}^N x_{ijk} = 1, \text{ para } i \in \{1, \dots, N\}; \quad (2.5)$$

$$\sum_{k=1}^K \sum_{i=0, i \neq j}^N x_{ijk} = 1, \text{ para } j \in \{1, \dots, N\}; \quad (2.6)$$

$$\sum_{i=1}^N m_i \sum_{j=0, j \neq i}^N x_{ijk} \leq q_k, \text{ para } k \in \{1, \dots, K\}; \quad (2.7)$$

$$\sum_{i=0}^N \sum_{j=0, j \neq i}^N x_{ijk} = 1 \quad (t_{ij} + f_i + w_i) \leq r_k, \text{ para } k \in \{1, \dots, K\}; \quad (2.8)$$

$$T_0 = w_0 = f_0 = 0; \quad (2.9)$$

$$\sum_{k=0}^K \sum_{i=0, i \neq j}^N x_{ijk} = 1 \quad (T_{ij} + t_{ij} + f_i + w_i) \leq T_j, \text{ para } i \in \{1, \dots, K\}; \quad (2.10)$$

$$e_i \leq (T_i + w_i) \leq l_i, \text{ para } i \in \{1, \dots, N\}; \quad (2.11)$$

$$\sum_{i=j}^n x_{jik} - \sum_{j=1}^n x_{ijk} = q_i; j = m + 1, \dots, n; \quad (2.12)$$

A função objetivo (2.1) representa o custo total da viagem dos veículos de todas as rotas. A restrição (2.2), garante que o número de viagens é K , selecionando K de arcos de saída do depósito ($i = 0$); A restrição (2.3), assegura que para cada veículo, é selecionada exatamente uma saída de um arco do depósito. Da mesma forma, a restrição (2.4) assegura que para cada veículo, existe exatamente um arco de entrada no nodo depósito ($j = 0$). As restrições (2.3 e

2.4) juntas, asseguram a viagem completa para cada veículo. A restrição (2.5) garante que, a partir de cada nó i apenas um arco para cada veículo dela emana. A restrição (2.6) garante que para cada nó j , apenas um veículo entra nele. As restrições (2.5 e 2.6) garantem que, cada veículo visita apenas um nó por vez. A restrição (2.7) garante que a soma total dos pedidos atribuída para cada veículo é inferior ou igual à sua capacidade. A restrição (2.8), assegura que o tempo total do percurso de cada veículo é inferior ou igual ao tempo máximo de atribuição da rota para esse veículo. A restrição (2.9) inicializa para zero, a hora de chegada, o tempo de espera e tempo de serviço de cada veículo quando está no depósito. O conjunto de restrições (2.10) garante de que o tempo de chegada de cada veículo no nó j é menor do que o tempo de chegada especificada para aquele nó. A restrição (2.11) garante de que a soma do tempo de chegada e o tempo de espera de cada veículo em cada nó i é maior ou igual ao início do tempo de chegada é menor um igual fim do tempo de chegada em cada nó i , já as restrições (2.9) e (2.11) definem as janelas de tempo (KUMAR; PANNEERSELVAM, 2012). A restrição (2.12), evita que existam sub rotas em ciclos.

A seguir, são mostrados os conceitos do PRV clássico e algumas variações do PRV.

2.2 Principais Variações do PRV

Este capítulo trata das principais variações do PRV, entre elas: o PRV capacitado (PRVC), com demanda estocástica (PRVDE), com múltipla viagem e problema de agendamento (PRVMVPA), com Frota Heterogênea (PRVFH), com coleta e entrega simultânea (PRVSCE), com múltiplos depósitos (PRVMD), com veículos aberto (PRVA) e com janelas de tempo (PRVJT), como estão descritos a seguir.

2.2.1 PRV Capacitado

Trata-se do Problema de Roteamento de Veículos Capacitado (PRVC) onde a capacidade do veículo é considerada. Dado um número fixo de veículos, com uma determinada capacidade, o PRVC consiste em encontrar rotas para os veículos, com um custo total mínimo, de tal modo que cada cliente pertença exatamente a um percurso, cada rota pertença a um depósito e que o total dos pedidos dos clientes associados a uma rota não exceda a capacidade do veículo (AUGERAT et al., 1998). A característica principal é que o total dos pedidos do cliente não deve exceder a capacidade do veículo.

2.2.2 PRV com Janelas de Tempo

No Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT), cada rota deve começar e terminar na janela de tempo associada ao depósito. Uma solução se torna inviável se um cliente é atendido após o limite superior da janela de tempo. Um veículo que chega antes

do limite inferior do período de tempo, causa um tempo de espera adicional na janela de tempo (DESROCHERS; DESROSIERS; SOLOMON, 1992). Normalmente, quando se lida com os problemas teóricos que envolvem janelas de tempo, é comum converter todas as medidas em unidades de tempo para fins de padronização e comparação de métodos (SUBRAMANIAN et al., 2012). A característica principal é de que o cliente deve ter o conjunto de pedidos atendidos no tempo estabelecido.

2.2.3 PRV Aberto

No Problema de Roteamento de Veículos Abertos (PRVA), um veículo não retorna para o depósito depois de atender o último cliente em uma rota, permanecendo neste. Cada rota é um conjunto de clientes visitados por um veículo. É necessário encontrar o número mínimo de veículos para atender a todos os clientes. Como exemplo, uma grande empresa contrata correios, que leva seus veículos até o depósito desta empresa. Pela manhã, carregam pacotes de carga e, em seguida, são feitas as entregas às residências. Os veículos dos correios não retornam ao depósito da empresa, após a sua última entrega do dia (LI; GOLDEN; WASIL, 2007). A característica principal é de que a rota inicia no depósito e termina no cliente.

2.2.4 PRV Múltiplos Depósitos

No Problema de Roteamento de Veículos com Múltiplos Depósitos (PRVMD), são fornecidos na entrada um conjunto de veículos, cada um tendo uma determinada capacidade, uma duração máxima, um conjunto de depósitos, um conjunto de clientes, uma demanda determinística e um tempo de serviço. O problema consiste em determinar as rotas a serem executadas para visitar cada cliente exatamente uma vez (ESCOBAR et al., 2014). As características principais são: o número de rotas associado a cada depósito, que não deve exceder um limite pré-estabelecido e cada rota deve começar e terminar no mesmo depósito.

2.2.5 PRV com Coleta e Entrega Simultânea

O Problema de Roteamento de Veículos com Coleta e Entrega Simultânea (PRVSCE) é uma variante do PRV, onde os clientes exigem simultaneamente o serviço de entrega e o serviço de coleta. Entregas são fornecidas a partir de um único depósito, no início do serviço do veículo, enquanto as cargas de coleta são levadas para o mesmo depósito na conclusão do serviço (MONTANÉ; GALVAO, 2006). A característica principal é de que os pedidos de um cliente podem pertencer a um fluxo de coleta, a um fluxo de entrega, ou aos dois fluxos combinados.

2.2.6 PRV com Frota Heterogênea

O Problema de Roteamento de Veículos com Frota Heterogênea (PRVFH) pode ser definido como uma frota composta por diferentes tipos de veículos. Considera-se uma unidade de distância, de um custo de viagem de cada veículo. Um conjunto de veículos (frota) está disponível no depósito. Cada um dos veículos tem características que correspondem a um tipo de veículo, com uma capacidade e custo fixo próprios. Para cada deslocamento do veículo, entre clientes ou entre depósitos e clientes, existe um custo de viagem do tipo variável, por unidade de distância, de um determinado tipo de veículo (SUBRAMANIAN et al., 2012). As características principais são de que a frota é composta por diferentes tipos de veículos e que cada modelo de veículo está associado a um custo fixo.

2.2.7 PRV com Múltipla Viagem e Problema de Agendamento

O Problema de Roteamento de Veículos com Múltipla Viagem e Problema de Agendamento (PRVMVPA) é semelhante ao PRV tradicional, porém possui algumas restrições mais específicas: cada dia um veículo pode fazer mais do que uma viagem; os clientes impõem janelas de tempo de entrega; os veículos têm capacidades diferentes tanto por volume quanto por peso; veículos podem ser contratados a qualquer momento, se a empresa não tem o suficiente; o acesso a alguns clientes é restrito em relação a determinados veículos; os horários dos motoristas devem respeitar o tempo máximo legal por dia, os intervalos de tempo legais e os tempos de descarga são levados em consideração (BRANDAO; MERCER, 1997). Característica principal deste problema é de que um veículo pode fazer mais do que uma viagem por dia.

2.2.8 PRV com Demanda Estocástica

O Problema de Roteamento de Veículos com Demanda Estocástica (PRVDE) considera a versão clássica do PRV, porém a demanda real é revelada, apenas quando o veículo chega ao cliente. A solução para este problema envolve a otimização dos horários do roteamento completo, com distância mínima de viagens, remuneração dos motoristas e da quantidade de veículos. O PRVDE difere de algumas abordagens determinísticas em que, quando alguns dados são aleatórios, e para algumas situações, não é possível exigir que todas as restrições sejam satisfeitas. Com o tempo limitado, uma possível ação corretiva, é aplicar uma penalidade quando a duração de uma rota excede o limite de tempo (TAN; CHEONG; GOH, 2007).

3 META-HEURÍSTICAS

Neste capítulo, são descritos alguns conceitos a respeito das meta-heurísticas. No primeiro momento é descrita uma breve introdução das meta-heurísticas. Depois, uma abordagem a respeito das heurísticas híbridas.

3.1 Introdução à Meta-Heurística

Segundo Glover e Laguna (2013), a denominação meta-heurística surgiu no mesmo período que foi introduzido o termo Pesquisa Tabu, e tem sido amplamente abordada na literatura, tanto nos títulos de estudos comparativos, como nos trabalhos de pesquisa. A meta-heurística refere-se à estratégia mestre que orienta e modifica outras heurísticas, que são normalmente geradas na busca de otimização local. A heurística guiada por uma estratégia pode possuir procedimentos de alto nível, ou pode assumir uma descrição de movimentos disponíveis. Isso ocorre, para transformar uma solução em outra, juntamente com uma regra de avaliação associada (GLOVER; LAGUNA, 2013).

Conforme Glover e Laguna (2013), é significativo o contraste entre a orientação da meta-heurística e a orientação na otimização baseado em busca local. Por muitos anos, a concepção preliminar de um procedimento heurístico era de prever, quer uma regra inteligente ou uma regra iterativa, em que há soluções imediatamente acessíveis para melhorar uma última solução encontrada. Essas heurísticas iterativas são muitas vezes referidas como métodos de descida, métodos de subida ou métodos de busca local. As Meta-heurísticas, nas formas modernas, baseiam-se em uma variedade de interpretações que constituem pesquisas inteligentes. Estas interpretações levam concepção de escolhas, que podem ser utilizadas para fins de classificação. No entanto, uma classificação rigorosa das diferentes meta-heurística é difícil e arriscada, porque os principais defensores de métodos alternativos diferem entre si, sobre a natureza essencial ou sobre os métodos que defendem (GLOVER; LAGUNA, 2013).

3.2 Heurística Híbrida

Uma Heurística Híbrida é uma abordagem para a resolução de problemas de otimização, que incorpora características de heurísticas diferentes. São conhecidas também por Heurísticas Compostas. Na teoria, a premissa para se construir um método híbrido é aproveitar as boas características de um método, e evitar as más características dos métodos originais. Reiter e Gutjahr (2012) propuseram, em seu trabalho, três hibridizações diferentes com algoritmos heurísticos: a primeira traz uma combinação sequencial; a segunda, uma combinação interativa. Na terceira, um otimizador multiobjetivo é executado, antes que o método de restrição adaptativa comece. Na combinação interativa, em cada iteração do método de restrição adaptativa, um único otimizador objetivo é chamado para gerar um candidato (REITER; GUTJAHR, 2012).

3.3 Algoritmo Genético (AG)

Este capítulo apresenta uma introdução ao Algoritmo Genético, conceitos sobre a geração da população, os métodos de seleção, alguns conceitos sobre a codificação dos cromossomos, as funções de aptidão, alguns métodos de cruzamento e recombinação e também sobre os métodos de mutação.

3.3.1 Introdução ao Algoritmo Genético

Segundo Baker e Ayechev (2003), o AG é uma técnica de otimização estocástica proposta por Holland (1975). A técnica foi inspirada no processo de evolução pela seleção natural, proposta por Darwin em 1859 na obra *A Origem das Espécies*. Uma população de soluções é mantida e um processo reprodutivo que permite que as soluções pais sejam selecionadas dentre as possíveis em uma população. As soluções descendentes são produzidas e herdadas algumas das características de cada um dos pais (BAKER; AYECHW, 2003).

O AG começa com um conjunto inicial de soluções possíveis, chamada população. Cada solução na população é chamada de cromossomo, o que representa um ponto no espaço de busca. Os cromossomos evoluem após sucessivas iterações, chamadas gerações. Durante cada geração, os cromossomos são avaliados utilizando algumas medidas de aptidão. Os cromossomos mais aptos têm maior probabilidade de serem selecionados para executar as operações genéticas, incluindo cruzamento e mutação. Na fase de passagem, o AG pode trocar porções dos pais selecionados, ou seja, dois cromossomos na população, para gerar uma prole. A operação de cruzamento acelera o processo para chegar a melhores soluções. Na fase de mutação, a operação de mutação mantém a diversidade na população para evitar ficar preso em um ótimo local. Uma nova geração é formada por selecionar alguns pais e alguns descendentes de acordo com os seus valores de aptidão, e rejeitando os outros para manter constante o tamanho da população. Após um número pré-determinado de gerações realizadas, o algoritmo converge para o melhor cromossomo, que representa a solução ótima ou pode ser uma solução quase ótima do problema (HO et al., 2008).

3.3.2 Geração da População Inicial

Segundo Baker e Ayechev (2003), a expectativa é que uma população inicial de soluções razoavelmente estruturada irá evoluir para soluções de alta qualidade em um número relativamente pequeno de gerações do AG. Normalmente, num AG, há uma população inicial, gerada de forma aleatória, que pode ou não conter necessariamente boas soluções candidatas. Uma versão modificada do conhecido Procedimento de Busca Adaptativa Gulosa e Randomizada (GRASP) pode ser utilizada para inicializar a população (MARINAKIS; MIGDALAS; PARDALOS, 2007).

Conforme Jeon, Leep e Shim (2007), o método de criação da população inicial, para o PRV, usa simultaneamente uma solução inicial, através de uma heurística e um método de geração aleatória. O método de geração aleatória das soluções criadas, a partir de números aleatórios e uma pesquisa global. A heurística também dá excelentes soluções iniciais que aumentam a possibilidade de que os mesmos veículos sejam atribuídos aos pontos de busca, dividindo a população em quatro áreas.

Conforme Ho et al. (2008), uma população inicial, também utilizada para o PRV, é construída de tal forma que cada indivíduo seja uma solução candidata viável, e desta maneira, todas as rotas na população inicial devem ser viáveis. No trabalho apresentado, parte da população é inicializada usando heurística, e parte é inicializada aleatoriamente. Um procedimento simples e uma heurística rápida são utilizados para distribuir todos os clientes nos veículos. Há um mecanismo utilizado para obter parte dos indivíduos da primeira geração, e com isso reduzir significativamente o tempo para o AG.

No trabalho de Baker e Ayechev (2003), a população é representada por cromossomos de vários custos (aptidão), que são classificados em uma ordem de custo crescente. Na população inicial, três cromossomos são gerados baseados em três heurísticas, e os outros são gerados como permutações dos cliente. A primeira heurística é um processo em duas fases simples, baseada no algoritmo de economia de um único para o PRV (CLARKE; WRIGHT, 1964). O *Saving* algoritmo corresponde a uma das mais famosas e eficazes heurísticas para o PRV, que é baseada na noção de economia. A estratégia é atribuir a cada cliente o depósito mais próximo. Na segunda heurística, cada cliente é atribuído ao depósito mais próximo. Em seguida, para cada depósito de rotas viáveis, são criadas por rotação de um raio centrado no depósito e, gradualmente, incluindo clientes num percurso do veículo até que a capacidade ou a distância da rota seja atingida. Uma nova rota é então iniciada e o processo é repetido até que todo o plano seja varrido. A terceira heurística é proposta da seguinte forma: Os clientes são divididos em principais clientes e clientes limítrofes, levando em conta a relação entre a distância ao seu depósito mais próximo e a distância para o segundo depósito mais próximo. Em seguida, os clientes são atribuídos aos seus depósitos mais próximos. Finalmente, os clientes limítrofes são inseridos, um por um, na ordem descendente do custo de oportunidade, onde o custo de oportunidade é a diferença entre a inserção do cliente em uma viagem começando no depósito mais próximo e a soma das distâncias entre o cliente inserido e outro cliente escolhido. A capacidade do veículo e a restrição da duração da rota devem ser satisfeitas (BAKER; AYECHHEW, 2003).

Conforme Ursani et al. (2011), um procedimento é concebido para construir blocos de rotas aleatórios e guardar durante as iterações anteriores na geração população. Isto ajuda a reduzir substancialmente o custo computacional. Isto porque, se gerar uma população aleatória, em seguida, o AG tem de reconstruir todos os blocos novamente, que foram construídos na iteração anterior. Este procedimento é utilizado para gerar a população inicial do AG. O procedimento mantém alguns pares de blocos de rotas selecionada, enquanto ainda ocorre a produção de uma população de diversos elementos de forma aleatória.

Este procedimento pode ser descrito nos passos como se segue (URSANI et al., 2011):

- Combinar o par da rota selecionada em uma única lista;
- Gerar aleatoriamente 3 pontos;
- Quebrar a lista em 3 pontos, ficando, assim, 4 blocos de construção;
- Inverter a ordem de cada um dos blocos de construção 4;
- Organizar em ordem aleatória os 4 blocos de construção.

3.3.3 Método de Seleção

Segundo Anbuudayasankar et al. (2012), a fase de reprodução favorece os melhores cromossomos, através do processo de seleção, ou seja, cromossomos com aptidão alta, são mais propensos a serem selecionados como pais. Segundo Baker e Ayechev (2003) a maneira mais clássica de reprodução ocorre com base em dois cromossomos pais da população atual. O processo de seleção é estocástico e um cromossomo com alta aptidão é mais provável de ser selecionado. Duas soluções pais são selecionadas a partir da população pelo método de torneio binário (*binary tournament method*) (BAKER; AYECHIEW, 2003). Assim, dois indivíduos são escolhidos na população de forma aleatória. Aquele com o melhor valor de aptidão é escolhido como o primeiro pai, da mesma forma, o processo é repetido para obter um segundo progenitor. Por fim, as descendências serão produzidas a partir de duas soluções pais, através do procedimento padrão cruzado.

3.3.4 Codificação

Segundo Potvin, Duhamel e Guertin (1996), os componentes principais da pesquisa genética são descritos resumidamente. O problema é o de maximizar uma função numérica f cujo domínio é um conjunto de números inteiros, cada cromossomo pode representar um número inteiro em notação de base 2. Neste caso, a aptidão de cada cromossomo é $f(x)$, em que x é o número inteiro codificado no cromossomo. No trabalho de Yücenur e Demirel (2011), são codificadas as variáveis de decisão em cadeias binárias. No código binário, cada cromossomo é definido com cadeias binárias formadas por 0 ou 1. Segundo Li, Leung e Tian (2012), cromossomo usado no AG proposto é codificado como permutação de todos os nós de clientes sem delimitadores de viagem, assim como um problema do caixeiro viajante visitando todos os nós do cliente. Cada cromossomo representa uma solução e possui um custo. O tipo de cromossomo binário não pode representar a solução PRVMD diretamente. O PRV necessita de um outro tipo de estrutura onde possa representar uma sequência de clientes. Algumas representações são visualizadas as sequências de depósitos e clientes, em outras os clientes são os índices e os genes do cromossomo são os veículos.

Conforme o trabalho de Ursani et al. (2011), a solução inicial é gerada pela escolha de uma ordem aleatória dos nós, formando uma lista de comprimento n , onde cada nó aparece apenas uma vez, porém o depósito não está incluído. A representação do cromossomo é da ordem de clientes sem divisores de rotas (URSANI et al., 2011).

Conforme o trabalho de Ho et al. (2008), a representação utilizada é a rota em que os clientes são listados, na ordem em que eles são visitados. Por exemplo, suponha que há seis clientes numerados 1 a 6. Se a representação da rota é (0, 2, 4, 1, 0, 3, 6, 5, 0), então duas rotas são obrigadas a servir todos estes seis clientes. Conforme é mostrado na Figura 1, na primeira rota um veículo começa a partir do depósito, em que está indicado como 0, viaja para clientes 2, 4, e, finalmente, ao cliente 1. Depois disso, o veículo retorna de volta para o depósito. Na segunda rota, o veículo começa com o cliente 3, então cliente 6 e, finalmente, ao cliente 5. Da mesma forma, o veículo viaja de volta para o depósito depois de servir os clientes. Note-se que cada cromossomo contém n ligações se houver n depósitos no PRVMD.

Figura 1: Representação do cromossomo

0	2	4	1	0	3	6	5	0
---	---	---	---	---	---	---	---	---

Fonte: (HO et al., 2008)

No trabalho de Ghoseiri e Ghannadpour (2010), que apresenta uma abordagem multiobjetivo com o AG, cada indivíduo é representado por apenas um cromossomo, que é uma cadeia de inteiros, cada um deles representando um cliente no PRV. Nesta representação, cada identificador de veículo (gene) com índice 0 representa no cromossomo um separador entre duas rotas diferentes, e uma série de identificadores de clientes representa a sequência de entregas, que deve abranger um veículo durante o seu percurso, conforme é mostrado na Figura 2.

Figura 2: Representação do cromossomo com a sequência da rota

0	4	5	2	0	10	3	1	0	7	8	9	0	6
---	---	---	---	---	----	---	---	---	---	---	---	---	---

Fonte: (GHOSEIRI; GHANNADPOUR, 2010)

No trabalho de Anbuudayasankar et al. (2012) foi utilizada a representação de cromossomo para a atribuição de clientes ao veículo. Cada gene representa um número do veículo e a posição de cada gene, representa os clientes em que o veículo é alocado. O exemplo tem dois veículos e quatro clientes e é codificado com que segue. Os clientes 1 e 4 são atribuídos ao veículo 1 e os clientes 3 e 2 são atribuídas ao veículo 2, conforme está representado na Figura 3. Este cromossomo, representa a alocação dos clientes para veículos, porém, não fornece a sequência de serviço pelo veículo. Assim, para tal representação do cromossomo, o autor utilizou a heurística de atribuição do veículo, para determinar a ordem do veículo atribuída para os clientes (ANBUUDAYASANKAR et al., 2012). Na Figura 3, não se consegue perceber visualmente qual a sequência correta clientes o veículo 1 deve seguir, por exemplo. A estrutura do cromossomo estabelece apenas a qual veículo um determinado cliente pertence.

Figura 3: Representação do cromossomo sem a sequência da rota

	1	2	2	1	Veículos
Cientes	1	2	3	4	

Fonte: (ANBUUDAYASANKAR et al., 2012)

Na representação mostrada na Figura 4, mostra na primeira linha os clientes e na segunda os veículos. O cromossomo indica que os clientes 2 e 5 serão visitados pelo veículo 1, os clientes 1, 6 e 7 pelo veículo 2, e os clientes e 3 e 4 pelo veículo 3. De forma equivalente, as variáveis de decisão são determinadas a partir da representação da composição do cliente com o veículo, como por exemplo C_{43} e $C_{12} = 1$ ou C_{51} C_{62} , C_{33} $C_{72} = 0$ (CHENG; WANG, 2009).

Figura 4: Representação do cromossomo

Cliente	1	2	3	4	5	6	7
Veículos	2	1	3	3	1	2	2

Fonte: (CHENG; WANG, 2009)

3.3.5 Função de Aptidão

A aptidão de cada solução deve estar relacionada com o valor da função objetivo, neste caso a distância total percorrida e o nível de qualquer violação de restrição. Análogo a processos biológicos, bons níveis de aptidão são mais propensos a sobreviver e reproduzir, com a expectativa de que os níveis de aptidão vão melhorar à medida que a população evolui (BAKER; AYECHEW, 2003).

No trabalho de Nguyen, Crainic e Toulouse (2014), para todas soluções, considera-se $c(s)$ o custo total de viagem das rotas, e $q(s)$, $d(s)$, $\omega(s)$ a violação da restrição de capacidade, duração da viagem e janela de tempo, respectivamente.

As soluções são avaliadas, de acordo com a função de aptidão ponderada, conforme a expressão (3.1), onde α , β e γ são parâmetros de penalidade ajustados dinamicamente, durante a pesquisa.

$$f(s) = c(s) + \alpha q(s) + \beta d(s) + \gamma w(s) \quad (3.1)$$

Já nos estudos de Ursani et al. (2011), a função de aptidão é descrita como o custo total (CT) das distâncias d entre os clientes n , na rota k , de acordo com a expressão (3.2).

$$CT = \sum_{k=1}^n d_k \quad (3.2)$$

3.3.6 Cruzamento e Recombinação

Uma vez ocorrida a fase de seleção, acontece a fase de cruzamento que cria dois novos herdeiros originários da troca genes e que são oriundos de dois cromossomos pais. Um cromossomo melhor possível de ser produzido, ocorre pela combinação genes, retirados de dois bons cromossomos. O *crossover* é aplicado para os veículos. O operador de cruzamento produz novas soluções potenciais, ajustando uma fração das soluções pais. Devido a isso, cada herdeiro detém sobre a algumas características de sua solução pai. Normalmente são utilizados um ou dois pontos de *crossover*, com base na probabilidade de cruzamento.

Segundo Jeon, Leep e Shim (2007), referente ao operador de cruzamento, primeiro seleciona um ou dois pontos de corte usando a probabilidade de cruzamento e gera dois filhos. O filho 1 recebe os valores do cromossomo pai 2, e o filho 2 recebe os valores do cromossomo pai 1. Os valores que não foram selecionados a partir de cromossomo pai 2 completam o filho 2. Os valores que não tenham sido selecionados a partir de um pai 1 completam o filho 1.

3.3.6.1 Operador de Cruzamento OX

Este operador recebe o cromossomo mãe e o cromossomo pai. Dois pontos são sorteados para montar uma subsequência interna no cromossomo pai. São considerados no cromossomo mãe, apenas os genes que não estão na subsequência do pai. Em seguida, é colocada no filho a subsequência retirada do pai, respeitando as mesmas posições dos genes no cromossomo pai. Por fim, nos espaços não ocupados no cromossomo filho, são acrescentados os genes que foram herdados da mãe, obedecendo à mesma ordem de posicionamento do cromossomo mãe. Com isso, está formada a cadeia completa do cromossomo filho. No trabalho publicado Li, Leung e Tian (2012), há um exemplo de operador de cruzamento OX. A Figura 5 mostra um exemplo do operador OX.

Figura 5: Operador OX

Pai	3	5	6	2	1	8	9	4	7
Mãe	9	6	3	1	4	7	8	2	5
Filho	9	3	6	2	1	8	4	7	5

Fonte: (LI; LEUNG; TIAN, 2012)

3.3.6.2 Operador de Cruzamento Mapeado Parcialmente PMX

Segundo Ursani et al. (2011), no PMX, há uma troca de sequência de genes escolhidos aleatoriamente entre dois cromossomos. Esta operação pode terminar na duplicação de genes e na omissão de outros. Pode-se substituir os genes duplicados, com os genes omitidos.

O PMX troca uma sequência de genes escolhidos aleatoriamente entre dois cromossomos.

Esta operação pode terminar na duplicação de genes e na omissão dos outros. Logo em seguida, são substituídos os genes duplicados, com os genes omitidos. No operador PMX, dois pontos aleatórios são escolhidos no cromossomo pai e no cromossomo mãe. Isso formará três subsequências nos cromossomos pai e no cromossomo mãe. As subsequências formadas pelos dois pontos escolhidos no pai, e entre dois pontos escolhidos na mãe formarão um mapeamento. O filho 1 recebe a sequência 1 do pai, a sequência 2 da mãe e a sequência 3 do pai. Já o filho 2 recebe a sequência 1 da mãe, a sequência 2 do pai e a sequência 3 da mãe. Esse cruzamento poderá ocorrer anomalias faltando alguns genes da sequência ou duplicando algum destes. Para que haja a correção dos genes duplicados, e dos omitidos, é utilizado um mapeamento entre os genes do pai e da mãe que corresponde as sequências centrais do pai e da mãe. Desta forma a sequência é ajustada devido ao mapeamento, e por posterior substituição dos genes duplicados e dos genes omitidos.

A Figura 6 ilustra um exemplo do operador de cruzamento PMX, onde o exemplo considera dois cromossomos pais: pai e mãe, e dois cromossomos filhos: filho 1 e filho 2.

Figura 6: Operador de Cruzamento PMX

Pai	1	4	7	9	6	3	2	5	8
Mãe	9	3	5	7	2	8	6	4	1
Filho 1	1	4	7	7	2	8	2	5	8
Filho 2	9	3	5	9	6	3	6	4	1
Filho 1	1	4	9	7	2	8	6	5	3
Filho 2	7	8	5	9	6	3	2	4	1

Fonte: (URSANI et al., 2011)

3.3.6.3 Cruzamento do Cromossomo de Duas Partes TCX

A Figura 7 mostra a representação do operador de cruzamento TCX. Um exemplo a seguir é utilizado para ilustrar o operador TCX, com 9 cidades e 3 rotas. Primeiro, um par de cromossomos mãe e pai são inicializados e o cromossomo mãe é usado como base para a geração de um filho. Depois, aleatoriamente, é selecionado um segmento do gene (sub-rota) da primeira parte do cromossomo mãe. Os genes selecionados são (7, 5) para a rota 1, (8, 4) para rota 2 e (1) para rota 3. Assim, são selecionados aleatoriamente são as quantidades de genes para três veículos para a mãe (2, 2, 1). Em seguida, a ordem dos genes remanescentes é classificada de acordo com as posições na primeira parte do cromossomo do pai.

Os genes restantes na primeira parte do cromossomo mãe são (9, 6, 2, 3), e a ordem desses genes é embaralhada ficando (2, 3, 9, 6) de acordo com a primeira parte do cromossomo do pai. Com base no número de genes não selecionados, é calculado um número aleatório uniforme entre 1 e o valor atual dos genes não selecionados. Isto servirá como base para, determinar o número de novos genes que serão adicionados para cada rota do cromossomo filho. Os genes não selecionados são (2, 3, 9, 6). É gerada a sequência de inteiros (2, 1, 1) de números ale-

atórios uniformes entre 1 e o número de genes atualmente não selecionados. Em seguida, no cromossomo filho, a rota 1 recebe genes (2, 3), a rota 2 recebe (9), e o rota 3 recebe (6). Por fim, é construída as duas partes do cromossomo filho. Na segunda parte do cromossomo filho é gerada a soma de cada posição dos vectores intermediários segunda parte de $(2, 2, 1) + (2, 1, 1) = (4, 3, 2)$ (YUAN et al., 2013). A composição final do cromossomo filho ocorre de forma alternada, ou seja, para o mesmo veículo primeiro os genes que vieram da mãe e depois os que vieram do pai, assim sucessivamente para todos os veículos, de forma que a sequência de clientes esteja completa.

Figura 7: Operador de Cruzamento TCX

		Depósito									D1	D2	D3
Passo		Cromossomo									V1	V2	V3
1	Mãe	9	7	5	6	2	8	4	3	1	5	2	2
	Pai	8	2	3	7	5	1	9	4	6	3	4	2
2	Mãe	9	7	5	6	2	8	4	3	1	5	2	2
											2	2	1
3	Pai	9	6	2	3	->	2	3	9	6			
											2	1	1
4	Filho	7	5	2	3	8	4	9	1	6	4	3	2

Fonte: (YUAN et al., 2013)

3.3.7 Mutação

Segundo Hwang (2002), o objetivo da mutação é realizar uma perturbação no cromossomo corrente, através da inserção de um novo gene que estava em outra posição. Existem vários operadores de mutação propostos para a representação de permutação, como troca recíproca, inversão, deslocamento e heurística.

No trabalho de Nguyen, Crainic e Toulouse (2014), o autor utiliza um operador de mutação simples de novos descendentes, em que dois genes são selecionados aleatoriamente e os seus valores são trocados. Assim, dois clientes escolhidos aleatoriamente são trocados entre os veículos, exceto nos casos em que os dois clientes fazem parte do mesmo veículo (BAKER; AYECHW, 2003).

Segundo Potvin, Duhamel e Guertin (1996), são propostos três operadores de mutação, baseados em ordens diferentes. Conforme Cheng e Wang (2009), um operador de mutação pode aleatoriamente escolher um cromossomo de um pool de soluções com uma probabilidade p_m , e aplicar a mutação baseada em ordem nos cromossomos escolhidos. O operador de mutação é aplicado a cada descendente gerado pelos operadores de cruzamento. A mutação consiste na alteração da atribuição padrão de alguns clientes, que são selecionados através de uma probabilidade. Um novo padrão é atribuído a cada indivíduo da população atual. Conforme Mirabi (2015), uma heurística de mutação é projetada com uma técnica de vizinhança para produzir

uma melhor prole. Um conjunto de cromossomos transformado a partir de um pai através da troca de alguns genes é considerado como a vizinhança. Apenas o melhor na vizinhança é usado para a descendência produzida pela mutação. No entanto, o objetivo da operação de mutação é promover a diversidade da população. Portanto, é necessário alterar a heurística de mutação original quando considerar o PRV para múltiplos depósitos. A modificação é que todos os vizinhos gerados são usados como descendência (MIRABI, 2015), (HO et al., 2008). O procedimento da operação mutação heurística é listado a seguir:

- Passo 1: Pegue três genes em um pai de forma aleatória;
- Passo 2: Gerar vizinhos para todas as permutações possíveis dos genes selecionados, e todos os vizinhos gerados são considerados como a descendência.

3.3.8 Troca

Conforme Yücenur e Demirel (2011), um método aleatório é utilizado para o operador de mutação. Neste método, dois genes vizinhos são escolhidos aleatoriamente e, em seguida, os locais são alterados (POTVIN; DUHAMEL; GUERTIN, 1996), (URSANI et al., 2011). A Figura 8 trata do método que altera aleatoriamente dois genes vizinhos, o cliente 3 na posição 2 é trocado pelo cliente 4 na posição 4. Assim, o cliente 3 passa a assumir a posição 2 e o cliente 4 passa a assumir a posição 4. Após as trocas os clientes 1, 2 e 5 permanecem nas posições originais.

Figura 8: Operador de Mutação Troca

Cromossomo	1	3	2	4	5
Cromossomo	1	4	2	3	5

Fonte: (POTVIN; DUHAMEL; GUERTIN, 1996)

3.3.8.1 Remove e Insere (*remove-and-reinsert RAR*)

Conforme os estudos de Potvin, Duhamel e Guertin (1996), um cliente é selecionado aleatoriamente em um cromossomo e é movido para outra posição. Os clientes localizados entre as duas posições são deslocados. Na Figura 9, o operador RAR é ilustrado. Neste exemplo, o cliente 3 é movido a partir da posição 2 para a posição 4. Após a troca o cliente 2 que estava na posição 3 passa a ocupar a posição 2 e o cliente 4 passa a ocupar a posição 3. Assim, os clientes 1 e 5 permanecem nas posições originais (POTVIN; DUHAMEL; GUERTIN, 1996).

Figura 9: Operador Remove e Insere

Cromossomo	1	3	2	4	5
Cromossomo	1	2	4	3	5

Fonte: (POTVIN; DUHAMEL;
GUERTIN, 1996)

3.3.8.2 Último será o Primeiro (*last-will-be-first LWBF*)

Este operador primeiro seleciona aleatoriamente um ponto de corte no cromossomo. Em seguida, a parte de extremidade é movida para frente do cromossomo. Por exemplo, o ponto de corte é selecionado, de forma aleatória, entre a segunda e terceira posições, e a sub-cadeia [2, 3, 4] é deslocada para frente da sub-cadeia [1, 5] (POTVIN; DUHAMEL; GUERTIN, 1996).

Na Figura 10, é mostrado um exemplo de utilização do operador LWBF.

Figura 10: Último será o primeiro LWBF

Cromossomo	1	5	2	3	4
Cromossomo	2	3	4	1	5

Fonte: (POTVIN; DUHAMEL;
GUERTIN, 1996)

3.3.8.3 Inversão (*inversion*)

O trabalho de Hwang (2002) aborda o operador de mutação denominado de Inversão. A ideia é descobrir dois pontos de corte dentro de um cromossomo de forma aleatória e, em seguida, inverter a *substring* entre essas duas posições. Na Figura 11, é mostrado um exemplo de utilização do operador LWBF.

Figura 11: Inversão

Cromossomo	1	5	2	7	8	6	4	3
Cromossomo	1	5	6	8	7	2	4	3

Fonte: (POTVIN; DUHAMEL;
GUERTIN, 1996)

3.4 Busca Tabu (BT)

Este capítulo traz uma breve introdução da meta-heurística Busca Tabu (BT), quanto ao espaço de busca, memória adaptativa, solução inicial, Lista Tabu (LT), tabu tenure, critérios de aspiração, critério de parada, listas de candidatos, estrutura de vizinhança, permutação, realocação, inserção, troca e deslocamento.

3.4.1 Introdução à Busca Tabu

Com base em alguns trabalhos anteriores, Fred Glover propôs, em 1986, uma nova abordagem, que chamou de Busca Tabu, no sentido de construir métodos para superar ótimos locais. Muitos elementos desta primeira proposta da BT, e alguns elementos das posteriores, foram introduzidos por Glover, em 1977, incluindo a memória de curto prazo e memória a longo prazo, para evitar a reversão de movimentos recentes. O princípio básico da BT é armazenar movimentos de não melhora, e evitar volta a soluções anteriormente visitados. Uma introdução à Busca Tabu retrata que soluções armazenadas em uma estrutura denominada LT, são impedidas de serem avaliadas. Segundo Glover e Taillard (1993), a BT pode ser associada a conceitos de inteligência artificial e, em algumas situações, pode não ser considerada propriamente uma heurística, mas sim, uma estratégia geral para orientar e controlar heurísticas internas (GLOVER; KOCHENBERGER, 2006). A seguir, são descritas algumas características da BT:

1. Espaço de Busca: no espaço de busca estão contidas todas as soluções possíveis; também conhecido como espaço de pesquisa. Nele, conforme o objetivo, é possível localizar uma solução em algum ponto extremo, mínimo ou máximo;
2. Memória Adaptativa: a memória adaptativa é um repositório de rotas escolhida entre as melhores soluções visitadas durante a pesquisa. Sua finalidade é fornecer novas soluções de partida para a BT, através da seleção e combinação de soluções extraídas da memória. Na primeira fase, a BT é utilizada para criar um número de soluções diferentes, que são armazenadas na memória adaptativa (BRÄYSY; GENDREAU, 2002);
3. Princípios Gerais: segundo Bräysy e Gendreau (2002), a ênfase na política de diversificação na BT, seja implementação determinística ou probabilística, deriva da suposição de que uma má escolha estratégica pode render mais informações do que uma boa escolha aleatória. A Diversificação, que integra os princípios básicos de pesquisa inteligente à exploração de características de boas soluções ao buscar novas regiões promissoras. Busca Tabu procura encontrar maneiras novas e mais eficazes de aproveitar os mecanismos associados com a memória adaptativa (BRÄYSY; GENDREAU, 2002);
4. Dimensões: as estruturas de memória na BT fazem referências a quatro dimensões principais, consistindo de recência, frequência, qualidade e influência. A dimensão por recência tem como objetivo evitar voltar à soluções recentemente visitadas, já por frequência está relacionada a memória de longa duração e que evita caminhos que apresentam determinadas características (BRÄYSY; GENDREAU, 2002). A dimensão da qualidade refere-se à capacidade de diferenciar o mérito de soluções visitadas durante a pesquisa. Neste contexto, a memória pode ser usada para identificar elementos que são comuns a boas soluções ou para caminhos que conduzem a essas soluções. Operacionalmente, a qualidade torna-se uma base para a aprendizagem baseada em incentivos, nas ações que levam

a boas soluções, e as sanções referentes às ações que conduzam a soluções pobres. A influência considera o impacto de uma determinada característica exerce sobre uma solução. A flexibilidade destas estruturas de memória permite que a busca seja guiada em um ambiente multiobjetivo, por exemplo (BRÄYSY; GENDREAU, 2002).

3.4.2 Solução Inicial

Conforme Li, Tian e Leung (2010), semelhante a outros algoritmos de busca local, a BT também precisa de uma solução inicial para começar a sua exploração, no espaço de soluções. Segundo Brandão (2004), qualquer algoritmo de BT requer uma solução inicial, que a partir dela, todo o processo de busca começa. Normalmente, a solução inicial é trivial, como atribuir cada cliente a uma rota, ou utilizar uma heurística rápida e bem conhecida. Conforme Brandão (2004), a solução inicial não tem tanta influência na qualidade da solução final, deixando todo o trabalho de melhoria para o algoritmo de BT. Brandão (2011) propôs um algoritmo para solução inicial, que mistura duas abordagens clássicas: vizinho mais próximo e inserção. Descreve que os métodos usados para gerar uma solução inicial devem ser simples e rápidos e que contemplem uma boa solução (BRANDÃO, 2011). Uma solução inicial é proposta utilizando o método do vizinho mais próximo, de maneira que, quando alguma restrição não for satisfeita, uma nova rota é iniciada (EUCHI; CHABCHOUB, 2010). A solução inicial é um estado no espaço de busca, e pode ser gerada de forma aleatória, porém, satisfazendo as restrições (CESCHIA; DI GASPERO; SCHAERF, 2011). Segundo o trabalho de Belhaiza, Hansen e Laporte (2014), a solução inicial considera que, durante a fase de inicialização, o melhor cliente para ser atribuído a um veículo é aquele que induz o menor aumento na soma total das distâncias entre os clientes. Uma vez que este procedimento pode gerar um conjunto de rotas que não servem a todos os clientes, ao final, são inseridos os clientes sem rota.

3.4.3 Lista Tabu

Movimentos tabus, são armazenados em uma memória de curto prazo, denominada de LT. Normalmente, apenas os movimentos são armazenados, ou seja, apenas alguns elementos de uma solução. Existem várias possibilidades relativas a quais informações armazenar. Pode-se armazenar soluções completas, mas isso requer uma grande quantidade de armazenamento e torna mais caro verificar se um movimento é tabu ou não, e raramente são usadas. Os tabus mais utilizados envolvem o armazenamento das últimas transformações efetuadas na solução corrente e que proíbe transformações reversas (GLOVER; KOCHENBERGER, 2006).

Segundo Glover e Kochenberger (2006), movimentos proibidos são usados para prevenir ciclos, que ocorrem com ótimos locais e movimentos de não melhora. Quando esta situação ocorre, algo precisa ser feito para evitar a pesquisa em regiões proibidas. Isto é possível por declarar que determinado movimento é tabu. No Problema de Roteamento de Veículos Capaci-

tados (PRVC), por exemplo, se o cliente acaba de ser movido, de uma rota para outra rota, pode ser declarado tabu, por um determinado número de iterações. Esse número é chamado de movimentos *tabu tenure*. Movimentos Tabus também são úteis para ajudar uma busca a afastar-se de regiões anteriormente visitadas no espaço de busca e, assim, realizar uma exploração mais extensa (LI; TIAN; LEUNG, 2010).

Movimentos de adiciona e retira (*Add and Drop*) estão relacionados à mudança do status do movimento na LT. Várias Listas Tabus podem ser utilizadas simultaneamente e são, por vezes aconselháveis. Por exemplo, se usada uma estrutura de vizinhança que contém tanto adiciona e retira e movimentos de troca, pode ser uma boa ideia manter uma LT separada para cada tipo de movimento. Listas tabus padrão são normalmente implementadas como listas circulares de comprimento fixo (GLOVER; KOCHENBERGER, 2006). Segundo Bräysy e Gendreau (2002), tabus de comprimento fixo podem nem sempre evitar ciclos, e alguns autores propuseram variações como o comprimento da LT durante a busca. Outra solução é gerar aleatoriamente o estado tabu de cada movimento dentro de algum intervalo especificado; o uso desta abordagem exige um esquema um pouco diferente para tabus que são armazenados como tags em uma matriz em que a entrada nessa matriz normalmente irá gravar o número da iteração para que o algoritmo controle o número de iterações que determina se o movimento ainda seja tabu.

No artigo de Wang, Li e Hu (2015), foi destacado que a LT é a memória mais utilizada a curto prazo para evitar um *loop* de volta para as soluções já visitadas. Um conjunto de atributos que caracterizam o movimento corrente gerado pela busca local é memorizado na LT com um tabu fixo de posse h (*tenure*). No processo, um movimento é armazenado como (nodo a , rota b) representando nodo a , que foi removido da rota b e, portanto, nodo a é proibido de ser reinserido da rota b , nas próximas h iterações.

3.4.4 Tabu Tenure

Tabu Tenure ocorre quando um número de iterações, onde um movimento que foi realizado sem melhora, é rastreado para não retornar a um status de não melhora da FO. Movimentos Tabus são usados para prevenir a ciclagem. A ideia chave aqui é que, quando esta situação ocorre, algo precisa ser feito para evitar a pesquisa, rastreando seus passos de onde veio. Isto é conseguido por declarar tabu (*disallowing*) movimentos que revertem o efeito de movimentos recentes. Por exemplo, no exemplo PRVC, se $v1$ é um cliente que acaba de ser movido de rota R1 até a R2, pode-se declarar tabu voltando $v1$ de R2 para R1 por algum número de iterações. Esse número é denominado Movimento Tabu Tenure (BRÄYSY; GENDREAU, 2002).

3.4.5 Critério de Aspiração

Critérios de aspiração são introduzidos na BT para determinar quando as regras de ativação tabu podem ser substituídas, eliminando assim uma classificação tabu de outra forma aplicada a um movimento. “Melhor-melhoria” e critério de “aspiração padrão” são exemplos simples e óbvios. O uso adequado de tais conceitos pode ser muito importante para permitir que um método BT possa atingir os seus melhores níveis de desempenho. As primeiras aplicações empregadas apenas como simples tipos de critério de aspiração, que consiste em remover uma classificação na tentativa de um movimento tabu baseia-se quando a solução do movimento é melhor do que o melhor obtido até o momento. Este critério continua a ser amplamente utilizado. No entanto, outros critérios de aspiração podem ser eficazes para a melhoria da pesquisa.

Em algumas situações, o status tabu é muito rigoroso para encontrar melhores soluções em que ele pode bloquear bons movimentos. A fim de resolver o problema, um critério de aspiração é invocado para revogar o estado de alguns movimentos, aceitando uma solução que seja tabu. A aspiração é realizada se um movimento tabu gera nova solução viável que é melhor do que a melhor solução viável atual. Os movimentos tabus podem ser substituídos se o critério de aspiração estiver satisfeito. (WANG; LI; HU, 2015).

3.4.6 Critério de Parada

Segundo Glover e Kochenberger (2006), são critérios que determinam a parada da execução do algoritmo BT. Um dos critérios, é o número máximo de iterações sem observar qualquer melhoria adicional no valor da FO. Em teoria, a pesquisa poderia continuar para sempre, a menos que o valor ideal do problema em questão seja conhecido previamente. Na prática, a busca tem que ser interrompida em algum ponto. Os critérios de parada mais comumente usados são: depois de um número fixo de iterações, um valor fixo de tempo de CPU, um número de iterações, sem uma melhoria no valor da FO ou quando o valor da FO atinge um limite pré-determinado. Nos esquemas complexos de tabu, a busca é normalmente encerrada após uma sequência de fases, a duração de cada fase é determinada por um dos critérios acima referido.

3.4.7 Lista de Candidatos

Listas de candidatos, são listas de soluções possíveis geradas para serem exploradas como a vizinhança de uma determinada soluções, com o objetivo de encontrar uma solução que melhore em relação ao valor da função objetivo. Conforme Glover e Laguna (2013), as listas de candidatos podem ser construídas a partir de regras relacionadas com o contexto e de estratégias gerais. São necessárias regras para a construção de listas de candidatos que são independentes do contexto. A eficácia de uma lista de candidatos, não deve ser medida em termos da redução

do esforço computacional em uma única iteração.

Em vez disso, uma medida preferível de desempenho para uma determinada lista de candidatos é a qualidade da melhor solução encontrada, dado um período de tempo em um computador. Por exemplo, uma lista de candidatos estratégica destina-se a substituir uma análise exaustiva na vizinhança, pode não resultar em mais iterações por unidade de tempo, mas pode necessitar de muito mais iterações para igualar a qualidade da solução do método original. Se a qualidade da melhor solução encontrada dentro de um limite de tempo desejável, ou através de uma série graduada de tais limites, não melhora, pode-se concluir que a estratégia da lista de candidatos não é eficaz (GLOVER; LAGUNA, 2013).

Eficiência e qualidade podem ser muito afetadas pela utilização de procedimentos inteligentes para isolar movimentos candidatos eficazes, ao invés de tentar avaliar cada movimento possível em uma vizinhança atual de alternativas. Isto é verdadeiro, quando uma vizinhança é grande ou cara para ser examinada. Os ganhos a serem alcançados por meio de listas de candidatos têm sido amplamente documentados, mas muitos estudos sobre BT ignoram a sua relevância (GLOVER; LAGUNA, 2013).

3.4.8 Estrutura de Vizinhança

Segundo Escobar et al. (2014), a estrutura de vizinhança é uma característica fundamental no desempenho de qualquer algoritmo de BT, pois determina a extensão e a qualidade do espaço de soluções explorado. Estruturas de vizinhança utilizadas na implementação proposta são baseadas em movimentos locais de troca de nodos, ou seja, intra-rota e inter-rota, troca (*swap*) e permutação (*exchange*). A estrutura de vizinhança define o mecanismo de geração de soluções de candidatos a partir da solução atual. No trabalho de Escobar et al. (2014), três operadores diferentes são utilizados para gerar soluções vizinhas: 2-Opt, permutação (*exchange*), e realocação (*relocate*). Os três operadores são aplicados à rota escolhida aleatoriamente da solução atual s para procurar potenciais melhorias. No trabalho de Escobar et al. (2014), destaca-se que a cada iteração na busca local são realizados mecanismos de troca em relação as soluções vizinhas, envolvendo rotas de veículos de coleta e veículos de entrega. Movimentos locais só são considerados se realmente forem soluções viáveis. Em relação à complexidade dos algoritmos, normalmente a vizinhanças é $O(n^2)$, e envolvem um número constante de trocas de nodos em uma rede representando os clientes. Tanto a inserção simples quanto a inserção dupla podem originar uma nova rota, no entanto, para as duas seguintes condições são observadas: (i) a rota de origem deve ter pelo menos 3 ou 4 clientes para a inserção simples e dupla, respectivamente; (I_i), (ii) o tipo de veículo da rota de destino, deve ser capaz de atender a demanda dos consumidores quando ocorrer o movimento, no caso da inserção do par, se a soma das demandas dos dois clientes não exceder a capacidade de qualquer um dos tipos de veículos (BRANDÃO, 2009, 2011).

Segue a descrição dos principais operadores para uma estrutura de vizinhança:

Permutação (*Exchange*)

Conforme o trabalho de Escobar et al. (2014), Li, Tian e Leung (2010), o operador de permutação consiste em trocar dois clientes pertencentes a duas rotas diferentes. Dois clientes consecutivos são transferidos de suas posições atuais para outras posições, mantendo a ligação dos nodos. Os clientes podem ser inseridos na mesma rota ou em uma rota diferente, bem como, atribuído ao mesmo depósito ou para um depósito diferente.

Inserção (*Insert*)

Um cliente é removido da sua posição atual e reinserido numa posição diferente na mesma rota ou em outra rota e atribuído ao mesmo depósito ou a um depósito diferente (ESCOBAR et al., 2014).

Troca (*Swap*)

Na troca, dois clientes, que pertencem à mesma rota ou rotas diferentes, designadas para o mesmo depósito ou para depósitos diferentes, são trocados (ESCOBAR et al., 2014). Este movimento envolve a realocação de dois clientes, i e j , que estão atualmente em rotas diferentes. O cliente i é removido do seu percurso, e inserido a uma posição na rota da qual j é removido. O cliente j é movido para uma posição na rota anteriormente pertencente a i . Todos os pares de clientes são considerados para esta operação dentro de uma iteração, e a melhor solução é escolhida. A complexidade computacional deste movimento é $O(N^4)$ (WASSAN; WASSAN; NAGY, 2008).

Deslocamento (*Shift*)

Este movimento implica passar um cliente i de uma rota para uma outra, com objetivo de melhorar a solução. Também é conhecido como operador $(I, 0)$. Todos os clientes são considerados para realizar o deslocamento e o movimento é implementado levando em conta a maior diminuição da distância total da rota. A complexidade computacional desta técnica é $O(N^2)$.

3.4.9 Diversificação e Intensificação na BT

Segundo Rochat e Taillard (1995), um princípio fundamental da BT é explorar a interação entre a diversificação e a intensificação. No trabalho de Crino et al. (2004), os processos da BT para intensificar ou diversificar a pesquisa são decisões baseadas em dados de pesquisa coletados e parâmetros de busca. A abordagem de Rochat e Taillard (1995), considera que a realização de uma exploração, baseia-se em perspectivas primárias. Baseia na ideia de avaliar um histórico de pesquisas que são mapeadas em probabilidades de seleção. Utilizando uma estrutura de vizinhança seleciona entre alternativas disponíveis, com uma atribuição de probabilidade que é tendenciosa para favorecer a escolha de avaliações mais elevadas.

A diversificação impulsiona a busca para examinar novas regiões. Conforme o trabalho de Wang, Li e Hu (2015), estratégias de diversificação orientam a busca para diferentes regiões menos exploradas.

Segundo os estudos de Brandão (2009, 2011), as principais fontes de diversificação são:

1. O uso de várias soluções iniciais e fases diferentes;
2. A oscilação estratégica;
3. A mudança do tabu tenure, da vizinhança e da frequência dos tipos de movimentos, durante a pesquisa.

Conforme Li, Leung e Tian (2012), para evitar o comportamento de estagnação, considera-se uma estratégia de diversificação, a reinicialização, com base em uma lista de elite, que armazena as melhores soluções durante o processo de iteração. Para grandes instâncias, onde o custo de CPU é alto, é necessário encontrar uma solução que tenha qualidade semelhante à nova solução corrente x . Para superar esta dificuldade, e com base na lista elite, seleciona-se aleatoriamente uma solução, e a pesquisa continua a partir desta solução.

Segundo Crino et al. (2004), a diversificação ocorre quando o processo de procura necessita explorar diferentes regiões daquela até o momento pesquisada. A diversificação normalmente transmite uma nova direção de pesquisa para uma partição diferente do espaço de solução. A diversificação afasta das atuais áreas de pesquisa local, onde movimentos tendem a ser sem melhora.

A intensificação concentra-se mais profundamente em regiões anteriormente encontradas para ser melhorada. A intensificação normalmente opera por recomeço, a partir de soluções de alta qualidade, ou modificando as regras de escolha para favorecer a inclusão dos atributos das soluções. A intensificação é alcançada principalmente quando o procedimento fica preso em um mínimo local, depois de iterações pré-definidas, a pesquisa irá reiniciar com a solução viável melhor visitada. Um limite é considerado para intensificar uma pesquisa, indicando que uma solução da próxima iteração, só será aceita, se for melhor do que a melhor solução distante viável encontrada. Segundo Li, Leung e Tian (2012), para a estratégia de intensificação pode ser construído um procedimento de pós-otimização, onde o objetivo é concentrar-se sobre a melhoria na distância total da rota, como avaliação em relação à melhor solução. Já Crino et al. (2004) defendem que a intensificação deve ser utilizada como uma estratégia para gerar boas soluções.

4 MODELO

O modelo gerado trata do PRV clássico considerando a restrição da capacidade do veículo, que determina que a totalidade dos pedidos dos clientes não ultrapasse a capacidade de carregamento do veículo. O modelo considera também que os pedidos dos clientes podem ser atendidos por diferentes depósitos, ou seja, o PRV Múltiplos Depósitos. Por fim, o modelo também considera que a frota que irá atender os pedidos dos clientes é composta por veículos de diferentes tamanhos caracterizando assim o PRV com Frota Heterogênea. Desta maneira o modelo é definido como o Problema de Roteamento de Veículo com Múltiplos Depósitos e Frota Heterogênea (MPRVDH). Uma rede é formada por clientes e depósitos, de maneira que os clientes possuem pedidos, e que os depósitos são responsáveis por atendê-los. São necessárias rotas de entrega para os veículos, que começam e terminam no depósito, de modo que todos os pedidos dos clientes sejam satisfeitos. O número de clientes varia de 50 a 360 e a empresa que realiza a distribuição possui mais de um depósito. Um custo é calculado pelas distâncias do depósito ao primeiro cliente, entre os clientes e do último cliente ao depósito. Somado a isto, ocorre o custo fixo do veículo, que varia de acordo com o tipo de veículo escolhido. O objetivo é otimizar o custo total da viagem que equivale a minimizar a distância total percorrida somado ao custo fixo do veículo. Considerando que o PRV é um problema de difícil resolução, conforme descrito no trabalho de Papadimitriou e Steiglitz (1982), foi proposto um modelo híbrido que contempla duas meta-heurísticas para resolver o problema proposto. Foram utilizadas o AG e a BT, sendo que estas podem ser utilizadas individualmente ou em conjunto. Quando as meta-heurísticas são utilizadas juntas, isto ocorre em sequência, primeiro o AG e depois a BT. As informações disponibilizadas pela implementação das meta-heurísticas, mostram os depósitos, as rotas, os clientes atendidos por cada veículo, o volume de carregamento do veículo, a distância total percorrida pelos veículos e suas rotas e o custo fixo dos veículos.

4.1 Premissas

As seguintes premissas e restrições devem ser respeitadas em relação ao modelo. No primeiro momento são mostradas aquelas relacionadas aos depósitos, depois aos veículos e por último aos clientes.

Os depósitos são onde estão armazenados os produtos que serão entregues aos clientes. A seguir são apresentadas as premissas vinculadas ao depósito:

1. O número de depósitos e suas localizações são previamente determinados.
2. Cada depósito possui um tamanho suficiente para armazenar todos os produtos solicitados pelos clientes.
3. As demandas são fornecidas por um único depósito.

4. Cada depósito possui condições de fornecer os mesmos produtos que os outros depósitos.
5. Um limite superior na quantidade de veículos em cada depósito é observado.

Os veículos são responsáveis por entregar os pedidos aos clientes. A seguir são apresentadas as premissas mais relacionadas ao veículo:

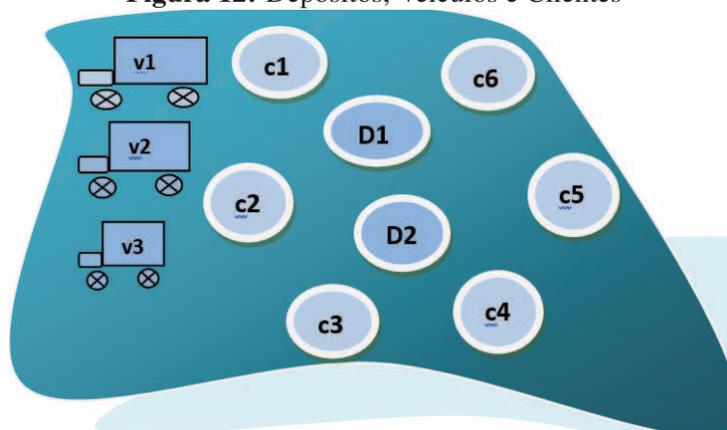
1. São definidos previamente, antes do roteamento, os tipos diferentes de veículos heterogêneos, em cada depósito.
2. Os veículos partem de qualquer um dos depósitos, e após entregar os pedidos aos clientes, retornam ao mesmo depósito.
3. As capacidades dos veículos são fornecidas anteriormente, para cada um dos tipos de veículo.
4. A soma dos pedidos de qualquer rota, não pode exceder a capacidade do veículo;
5. Um veículo possui apenas uma rota.

Os clientes possuem suas demandas e necessitam que sejam supridas, e estão espalhados geograficamente. A seguir são apresentadas as premissas relacionadas ao cliente:

1. A cada cliente é atribuída uma demanda não-negativa que são os pedidos;
2. Cada cliente é visitado apenas uma vez e exatamente por um veículo.
3. Cada veículo deixa este cliente após atendê-lo, garantindo a continuidade do caminho do veículo.

A Figura 12 mostra os clientes, os depósitos e os três tipos diferentes de veículos.

Figura 12: Depósitos, Veículos e Clientes



A seguir é apresentada a formulação matemática correspondente ao modelo proposto.

4.2 Formulação Matemática

Nesta secção é proposta, a formulação matemática referente ao Modelo do Problema de Roteamento de Veículo, considerando Múltiplos Depósitos e Frota Heterogênea. Neste modelo, diferentes clientes estão dispostos em localidades e possuem suas demandas representadas por pedidos. O fornecedor possui diferentes depósitos distribuídos de forma regional, de maneira que, cada depósito possui os mesmos produtos a serem distribuídos aos clientes. Os clientes concorrem entre si para utilização dos veículos (AFSHAR-NADJAFI; AFSHAR-NADJAFI, 2014).

Os pedidos de cada cliente são fixos, os parâmetros e as variáveis de decisão são os seguintes:

A = conjunto de arcos de um grafo que representa rotas potenciais entre os locais;

N = conjunto de nodos de um grafo que representa os clientes e os depósitos onde $(i, j \in N)$;

W = conjunto de depósitos $W \subset N$ onde $(i \in j = 1, \dots, m)$;

V = conjunto de clientes em que $V \subset N$, $W \cap V = \emptyset$; $W \cup V = N$ onde $(i \in j = m+1, \dots, n)$;

F = representa os veículos heterogêneos onde $(k = 1, \dots, K)$;

P = número de tipos de veículos;

Q = número máximo de veículos em cada depósito depois do roteamento;

q_i = pedidos do cliente i ;

C_k = capacidade do veículo k ;

cf_k = custo fixo do veículo k ;

d_{ij} = distância entre as locais i e j ;

c_{ijk} = custo de viagem entre os locais i e j com o veículo k ;

Z = função objetivo;

$x_{ijk} = 1$, se o veículo k , viaja entre os locais i e j e 0, caso contrário; (variável binária).

Função objetivo: A FO representa o custo total das viagens entre os depósitos e os clientes e o custo fixo total dos veículos conforme a Fórmula 4.1. Nesta função é considerado o somatório do custo c da viagem pelos veículos. Considera-se: $W \cup V = N$; além disso, cada veículo heterogêneo possui um custo fixo Cf . A variável X_{ij} indica se o veículo é utilizado ou não, para uma determinada viagem de um depósito a um cliente, ou de um cliente a outro cliente.

$$MinZ = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^K c_{ij}^k x_{ijk} + \sum_{i=1}^m \sum_{j=m+1}^n \sum_{k=1}^K cf_k x_{ijk}; \quad (4.1)$$

Cliente visitado uma vez: as restrições 4.2 e 4.3 garantem que o cliente j no depósito m seja visitado pelo veículo k , apenas uma vez.

$$\sum_{i=1}^n \sum_{k=1}^K x_{ijk} = 1; j = m + 1, \dots, n; \quad (4.2)$$

$$\sum_{i=1}^m \sum_{j=m+1}^n x_{ijk} \leq 1; k = 1, \dots, K; \quad (4.3)$$

Permanece no Depósito: as restrições 4.4 determinam que os veículos permaneçam no depósito se não forem usados.

$$\sum_{i=m+1}^n \sum_{j=1}^m x_{ijk} \leq 1; k = 1, \dots, K; \quad (4.4)$$

Limite de veículo por depósito: a restrição 4.5 garante um limite de um número máximo de veículos, para cada depósito, após o roteamento;

$$\sum_{i=m+1}^n \sum_{k=1}^K x_{ijk} \leq Q; j = 1, \dots, m; \quad (4.5)$$

Veículo não termina no cliente: A restrição 4.6 impede que o veículo termine no local do cliente.

$$\sum_{i=1}^n x_{irk} = \sum_{j=1}^n x_{rjk}; k = 1, \dots, K; \text{ onde } : r = m + 1, \dots, n; \quad (4.6)$$

Capacidade máxima: A restrição 4.7 determina que a capacidade máxima do veículo não seja violada.

$$\sum_{i=m+1}^n q_i \sum_{j=1}^n x_{ijk} \leq C_k; k = 1, \dots, K; \quad (4.7)$$

O deslocamento do veículo entre dois locais: especifica que as variáveis de decisão x_{ijk} são binárias, ou seja, indica: 1 se o veículo k viaja entre os locais i e j , e 0, o caso contrário.

$$x_{ijk} \in \{0, 1\}; k = 1, \dots, K; e(i, j) \in A; \quad (4.8)$$

Evita Sub rotas: A restrição que consiste na eliminação de sub rotas é expressa na equação 4.9, onde S :

$$S = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^K x_{ijk} \leq |B| - 1 \text{ para } B \subseteq V / \{0\}; |B| \geq 2; \quad (4.9)$$

B é um subconjunto de clientes do conjunto V .

Para implementar este modelo matemático foi elaborado uma arquitetura computacional, que terá a função de encontrar boas e viáveis soluções para o problema proposto, assim como, controlar as restrições apresentadas nesta secção. Na próxima secção é apresentada a arquitetura para o modelo proposto.

4.3 Arquitetura do Modelo Híbrido (MPRVDH)

Foi construída uma arquitetura para realizar a implementação do modelo que trata o PRV abordado neste trabalho. A arquitetura está estruturada cinco módulos que terão a função de ler diferentes arquivos referentes a problemas teste, montar uma rede que representa a conexão entre os clientes e depósitos, construir uma solução inicial e o módulo principal que fará a execução dos algoritmos AG, BT e AH e exibirá os resultados. Seguem os módulos que compõem a arquitetura.

- Montar a Rede;
- Geração da Solução Inicial;
- Algoritmo Híbrido (AH).
- Algoritmo Genético (AG).
- Algoritmo Busca Tabu (BT).

Em seguida, é realizada a montagem da rede, que é composta pelos depósitos e clientes e as distâncias entre eles. Uma estrutura de dados armazena as demandas dos clientes e outra armazena as informações do custo e capacidade dos veículos. O módulo "Geração da Solução Inicial" constrói uma solução inicial que servirá como a primeira solução para a BT ou para a geração da população no AG. A saída deste módulo representa uma sequência de clientes, os veículos que irão atender seus pedidos e os depósitos de onde sairão os veículos.

4.3.1 Montar a Rede

O módulo "Montar a Rede", tem a responsabilidade de realizar a entrada de dados no sistema, e ocorre em dois procedimentos distintos: o primeiro faz a leitura dos arquivos e o segundo constrói a rede armazenando os dados lidos em estruturas de dados em memória. Assim, é realizada a leitura de arquivos textos que contém informações e parâmetros importantes para a sequência dos outros módulos. Os arquivos textos são problemas testes de autores conhecidos que já trataram o PRV.

4.3.1.1 Ler Arquivos de Entrada

Neste módulo é feita a leitura dos arquivos de dados e dos parâmetros como: a quantidade de clientes, o número de depósitos, número de veículos, a capacidade dos veículos e os tipos de veículos. Estes são referentes aos problemas teste, de trabalhos já publicados. Além disso, traz os dados referentes à localização dos clientes, as coordenadas geográficas que correspondem à localização dos clientes. No arquivo, estão informações como o número dos clientes, os

números dos depósitos e as coordenadas geográficas. Os arquivos também contêm os pedidos de cada cliente e da capacidade dos veículos. A Figura 13 mostra um exemplo de um arquivo de um problema teste.

Na primeira linha estão descritos quatro valores {tipo, número de veículos, número de clientes, número de depósitos}. Na segunda linha são lidos dois valores {duração máxima da rota, capacidade máxima do veículo}, neste caso, a duração não foi informada. Nas linhas subsequentes vêm as informações dos clientes e dos depósitos, no arquivo original, são mostrados vários valores que não foram utilizados neste trabalho. Os dados utilizados aqui são {número do cliente ou depósito, coordenada X, coordenada Y e a demanda do cliente}. Nas duas últimas linhas, as informações são referentes aos depósitos, que contém as coordenadas X e Y, porém a demanda zero.

Figura 13: Arquivo do Problema Teste

2	2	6	1
0	90		
1	37	52	7
2	49	49	30
3	52	64	16
4	20	26	9
5	40	30	21
6	21	47	15
7	20	20	0
8	30	30	0

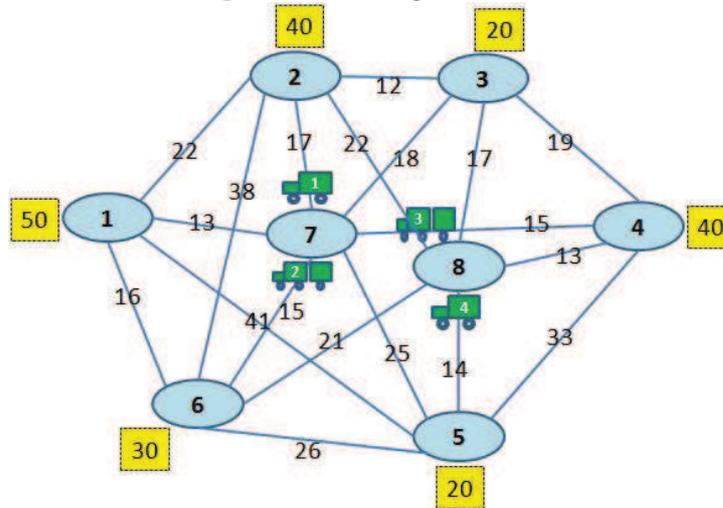
4.3.1.2 Popular as estruturas de dados

Após a leitura dos arquivos a distância Euclidiana é calculada, considerando cada par de vértices $A = (a_x, a_y)$ $B = (b_x, b_y)$, ela é calculada pela expressão (4.10).

$$D = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2} \quad (4.10)$$

A rede é representada por um grafo não-direcionado na qual os vértices são os depósitos ou clientes e há um custo associado a cada vértice que é a distância D . A cada cliente possui uma demanda e, cada depósito tem associado a si, um ou mais veículos, sendo que cada veículo possui um custo fixo. Todas as arestas da rede se conectam, ou seja, os depósitos estão ligados a todos os clientes e os clientes estão conectados aos outros clientes. Na Figura 14, está uma ilustração da rede. As arestas de 1 a 6 representam os clientes com suas demandas e as arestas 7 e 8 representam os depósitos. O depósito 7 tem os veículos 1 e 2 e o depósito 8 tem os veículos 3 e 4. Pode se dizer por exemplo, que entre os clientes 1 e 2 a distância é de 22 e entre os clientes 4 e 5 a distância é de 33. É possível ver que entre o depósito 7 e o cliente 3 a distância é 18. Por fim, que o cliente 4 tem uma demanda 40 e que os clientes 3 e 5 têm a demanda 20.

Figura 14: Montagem da rede



Após a leitura dos arquivos os valores das coordenadas são utilizados para calcular a distância entre os clientes e clientes e depósitos. Então estes dados são armazenados em uma matriz de distâncias onde linhas e colunas representam os clientes e os depósitos e as células representam as distâncias. Na Figura 15, está a representação da matriz de distâncias. A matriz é considerada simétrica de forma que $M[1][2]$ é igual a $M[2][1]$, ou seja, a distância entre o cliente 2 para o cliente 1 é a mesma do cliente 1 para o cliente 2.

Figura 15: Representação da Rede

	1	2	3	4	5	6	7	8
1	0	22	33	40	41	16	13	22
2	22	0	12	28	14	38	17	22
3	33	12	0	19	33	23	18	17
4	40	28	19	0	10	34	15	13
5	41	14	33	33	0	26	25	14
6	16	38	23	34	26	0	15	21
7	13	17	18	15	25	15	0	-
8	31	25	17	13	14	21	-	0

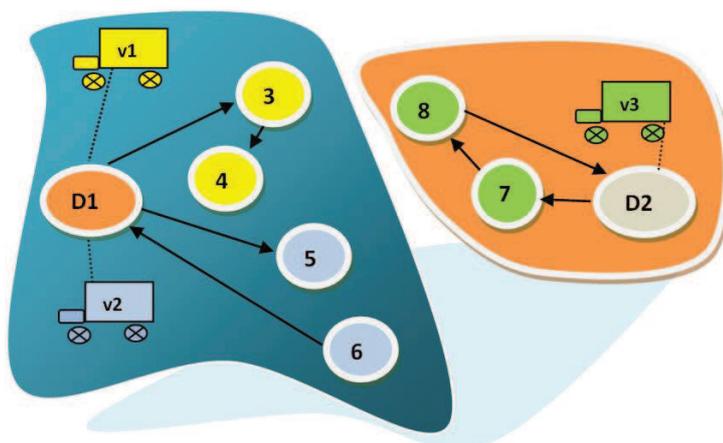
4.3.2 Geração da Solução Inicial

Neste procedimento, após a leitura dos arquivos que contêm as coordenadas e os pedidos dos clientes, é construída uma solução inicial. Para cada cliente é verificada, em relação aos demais, a menor distância, e assim sucessivamente até o último cliente estar associado a um outro. Durante este processo um cliente é atendido por um veículo que pertence ao depósito mais próximo, até que o número máximo de veículos por depósito seja atingido. Aqui, duas restrições são verificadas: o respeito à capacidade dos veículos e o limite de veículos por depósito. Cada vez que um pedido do cliente, mais a soma dos pedidos já associados a um determinado veículo,

ultrapassar a sua capacidade, o cliente é atendido por um novo veículo. Cada vez que atingir o limite de veículos por depósito, o cliente é associado a um veículo cujo o depósito não tenha o limite ultrapassado.

No final do procedimento é armazenada, em uma estrutura de dados, a sequência dos clientes, os veículos e os depósitos correspondentes a eles. O procedimento controla para que a solução seja viável, ou seja, todos os clientes sejam atendidos pelos veículos respeitando sua capacidade de carregamento e os depósitos com o número máximo de veículos. O objetivo é gerar uma solução boa para que nos procedimentos posteriores, que envolvem o AG e o BT, a melhora não seja tão custosa, do ponto de vista de esforço computacional, caso fossem executados sem uma solução inicial. A Figura 16 mostra uma solução inicial com os depósitos os veículos e os clientes. Os clientes 3 e 4 são atendidos pelo veículo $v1$ e pelo depósito $D1$, já os clientes 5 e 6 são atendidos pelo veículo $v2$ e também pelo depósito $D1$. Os clientes 7 e 8 são atendidos pelo veículo $v3$ e pelo depósito $D2$.

Figura 16: Solução Inicial



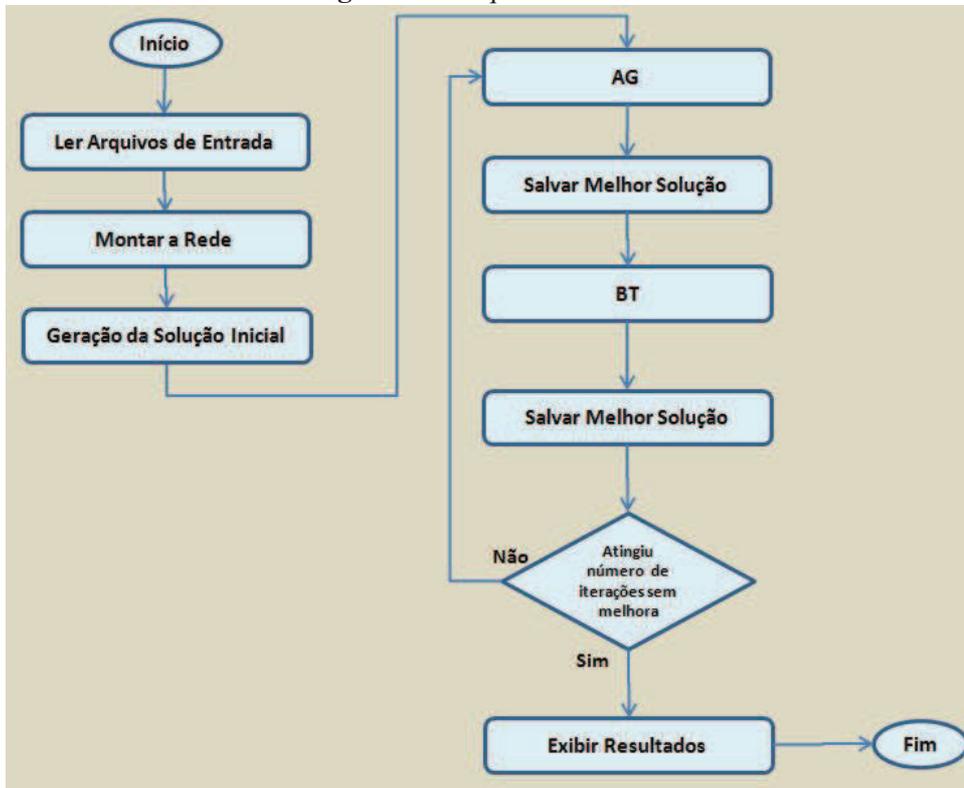
4.3.3 Algoritmo Híbrido AH

No módulo "AH" acontece o processamento das heurísticas implementadas, o AG e a BT. A solução inicial é utilizada na geração da população no AG e armazenada como a melhor solução do AH, na primeira iteração. A melhor solução, da saída do AG entra como solução inicial para BT, que devolve a melhor solução encontrada após a execução dela.

A melhor solução resultante do AG é armazenada como a melhor solução do AH, caso melhore em relação a melhor solução anterior. O módulo AG pode ser executado separadamente. Desta forma, recebe, a solução inicial, gera a população, executa os operadores de cruzamento e mutação e a ABL. Repete estes procedimentos tantas vezes conforme o número de gerações. O módulo "BT" pode ser executado separadamente também. Recebe a solução inicial, gera a vizinhança de acordo com os movimentos de troca, retira e insere e deslocamento. Isso ocorre tantas vezes, na busca de uma melhor solução, conforme o parâmetro que corresponde a um

número de iterações sem a melhora da solução.

Figura 17: Arquitetura do AH



A Figura 17 mostra a seqüência de módulos e funções que compõe a arquitetura do AH. Primeiro é executado o AG e depois a BT.

4.3.3.1 Possíveis execuções

O algoritmo, foi construído de maneira a ser executado de três formas diferentes conforme as variáveis de decisão que representam o AH, AG e BT. Todos os três tipos de processamento passam pela "Montagem da Rede", pela "Geração da Solução Inicial", "Salvar Melhor Solução" e pelo "Exibir Resultados". A primeira possibilidade é a execução do AG separado. A segunda possibilidade é a execução da BT separada. Por fim, a terceira consiste em concatenar as duas meta-heurísticas, passando primeiro pelo AG e depois pela BT em sequencia.

4.3.3.2 Salvar Melhor Solução

1. Local

No AG, este procedimento escolhe a solução elite com a função de aptidão melhor a cada geração da população. Na BT, guarda a melhor solução ao término de cada avaliação de uma geração da vizinhança;

2. Global

No AG é armazenada a melhor solução, entre todas as soluções locais encontradas, independente da população que foi gerada. Na BT é armazenada a melhor solução, entre todas as soluções locais encontradas, independente da vizinhança gerada. No AH é armazenada sempre a melhor solução entre as melhores soluções do AG ou BT.

4.3.3.3 Exibir Resultados

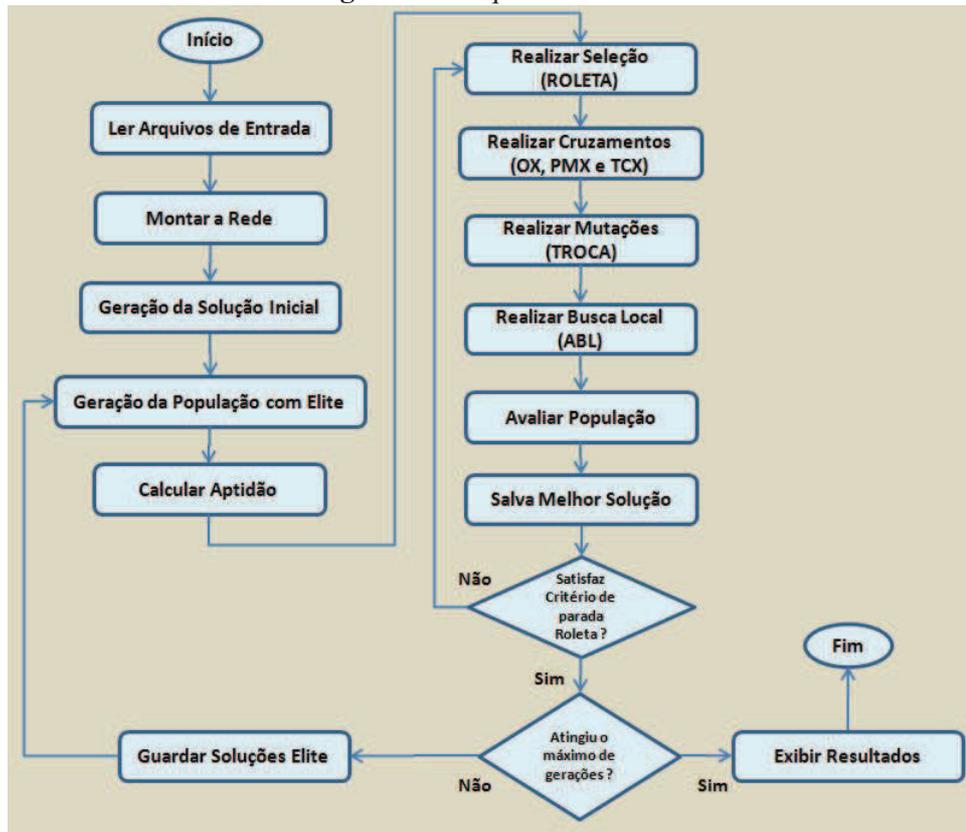
Após terminarem as iterações, o programa exibe as informações referentes a rotas, veículos, clientes e distância. As informações finais exibidas são: a distância total, distância por rota, clientes por rota e a capacidade do veículo por rota, os clientes, os depósitos e os veículos. Um procedimento irá gerar os relatórios correspondentes às execuções do algoritmo. Além disso, o algoritmo também é responsável por exibir informações operacionais como, a matriz de distâncias, número de iterações sem melhora no AH e na BT.

4.3.4 Execução somente do AG

O procedimento recebe a solução inicial e devolve como saída a melhor solução encontrada. Este procedimento é executado individualmente ou quando o AH é executado. Os procedimentos executados neste módulo são: Geração da População, Calcular Aptidão, Realizar Seleção, Realizar Cruzamentos, Realizar Mutações, Realizar à Busca Local, Avaliar População, Guardar Soluções Elite, Salvar Melhor Solução Local, Salvar Melhor Solução Global, Geração da População com Soluções Elite,

Conforme a Figura 18, o procedimento recebe uma solução, denominada solução inicial, e as soluções elites geradas a partir desta solução inicial. A cada geração, primeiro é construída uma população nova, respeitando sempre a população elite, que são as melhores soluções armazenadas nas gerações anteriores. A estratégia de seleção utilizada é a roleta. Neste procedimento, um parâmetro define a quantidade de vezes que a população é explorada. A função utiliza a taxa de cruzamento, taxa de mutação e taxa para execução do Algoritmo de Busca Local (ABL). Caso a probabilidade de cruzamento sorteada seja menor que o parâmetro taxa de cruzamento, são escolhidas duas soluções pais. Conforme os parâmetros de execução, o sistema escolhe um operador de cruzamento que pode ser o PMX, o TCX e o OX. Caso a probabilidade de mutação sorteada seja menor que o parâmetro taxa de mutação, o operador de mutação *swap* é executado. Desta forma, o mesmo ocorre com a avaliação de executar o ABL, que deve ser menor do que o parâmetro de execução do ABL. Os operadores de cruzamento utilizam um pai e uma mãe e geram um filho 1 e um filho 2. Já os operadores de mutação e o ABL, podem transformar o filho 1 e o filho 2 em novos cromossomos, caso tenham o valor da função objetivo melhorado. Conforme o resultado do filho 1 em relação ao pai ou o filho 2 em relação a mãe, as soluções pai e mãe são substituídas pelas soluções filhas. Desta forma, se o valor da função

Figura 18: Arquitetura do AG



objetivo dos filhos for menor que o valor da função objetivo dos pais, os pais são substituídos na população pela nova solução dos filhos. Na sequência é feita a avaliação da população onde são armazenadas nas primeiras posições as melhores soluções das gerações ocorridas. Ao final da geração o sistema armazena a melhor solução encontrada nesta população e atualiza a população elite. Para determinar o tamanho da população elite, um parâmetro é utilizado.

4.3.4.1 Geração da População

O objetivo é construir uma população que tenha N cromossomos como solução de elite, e os restantes até o tamanho da população, gerados aleatoriamente. A solução elite tem como base a solução inicial, que na primeira vez, como não existe ainda soluções elites, apenas a solução inicial é elite e o restante é gerado aleatoriamente para $N - E$ elementos da população elite. Já nas interações seguintes o algoritmo armazena como elite as E melhores soluções conhecidas da população. Para o restante das soluções não elite da população, é realizada um sorteio aleatório da sequência dos clientes. O procedimento considera uma estrutura com os clientes, os veículos e os depósitos. A restrição de carga do veículo é respeitada, ou seja, o pedido do cliente mais a soma dos pedidos dos clientes que estão associados a um veículo, não podem ultrapassar a capacidade deste. Além disso, também é respeitado a restrição do número de veículos por depósito.

Na Figura 19, está mostrada a representação do cromossomo referente ao modelo do PRV com Múltiplos Depósitos e com Frota Heterogênea. A primeira linha da matriz representa os clientes que deverão ser atendidos, a segunda referente ao veículo, a terceira os depósitos. Esta codificação é semelhante ao trabalho apresentado por Cheng e Wang (2009). A representação do cromossomo ocorre da seguinte forma: O cliente 4 será atendido pelo veículo 1 e será atendido pelo depósito 1; os clientes 5, 6 e 7 serão atendidos pelo veículo 2 e pelo depósito 2; os clientes 9 e 10 serão atendidos pelos veículos 3 e 4 e pelo depósito 3.

Figura 19: Representação do cromossomo

Clientes	1	2	3	4	5	6	7	9	10
Veículos	1	1	1	1	2	2	2	3	4
Depósito	1	1	1	1	2	2	2	3	3

4.3.4.2 Calcular Aptidão

Este procedimento é responsável pelo cálculo do valor da FO, para cada solução gerada. Por parâmetro é recebida uma estrutura que representa um cromossomo com os clientes, veículos e depósitos. O cálculo do valor FO é representado pela distância total da rota de cada veículo, que considera as distâncias entre os clientes, a distância entre o depósito e o primeiro cliente e mais a distância do último cliente e o depósito. O valor da FO considera a soma total das distâncias e mais a soma total dos custos fixo para cada veículo utilizado na solução. A Figura 20 mostra uma estrutura que armazena a solução, para ser feito o cálculo da aptidão. Os clientes 1, 2, 3 pertencem ao depósito 1 e ao veículo 1, por exemplo. Para calcular o custo total do veículo 1 são consideradas as distâncias que partem do depósito 1 para o cliente 1, do cliente 1 para o cliente 2, do cliente 2 para o cliente 3, do cliente 3 para o cliente 4 e do cliente 4 para o depósito 1. Todos os outros veículos têm o custo total calculado, seguindo a mesma lógica aplicada ao veículo 1. A coluna Tot.Pedidos é utilizada no controle das restrições da capacidade dos veículos.

Figura 20: Estrutura para o Cálculo da Aptidão

Depósito	Veículo	Clientes				Cf	Tot. Custo	Tot.Pedidos
1	1	1	2	3	4	15	80,30	72
1	2	5	6	7		15	60,40	60
2	3	9				15	30,58	20
2	4	10				15	25,20	15
Custo Total Solução:							196,48	
<i>CF= Custo Fixo do Veículo</i>								
<i>Tot. Custo= Cf+Total das distâncias entre clientes e depósito</i>								
<i>Tot. Pedidos= Total dos pedidos dos clientes para um veículo</i>								

4.3.4.3 Realizar Seleção

Este procedimento recebe a população e retorna uma solução pai e uma solução mãe, que são utilizadas na sequência dos operadores do algoritmo genético. Uma estrutura com o tamanho da população, armazena os dados que são utilizados para a execução da seleção. A estrutura (matriz) armazena o número da solução h , o valor da função objetivo Z , o peso PB e o peso acumulado Q . O procedimento realiza a soma de todas as aptidões TA . A matriz é ordenada, de forma decrescente, pelo valor da aptidão de cada indivíduo. Em seguida, é realizada a soma do peso acumulado, que é peso do indivíduo com o peso das soluções anteriores. O procedimento realiza primeiro o sorteio para selecionar a solução mãe e depois para a solução pai, contudo é feita uma consistência para que o pai e a mãe sejam diferentes. Ao final, o procedimento retorna os índices identificadores da solução mãe e da solução pai.

Selecionar indivíduos de uma população se realiza da seguinte forma: calcular o peso de cada indivíduo em relação à população, com base nos valores de aptidão. A abordagem utilizada é a Roda de Roleta (RR), semelhante ao trabalho de Yücenur e Demirel (2011) e Mirabi (2015) que considera os seguintes passos:

- Calcular a aptidão, para cada indivíduo C ;
- Calcular a aptidão total TA da população P ;
- Calcular o peso de seleção para cada indivíduo:
- Calcular o peso cumulativo para cada indivíduo:
- Gerar um número aleatório r menor que o valor acumulado $C1$ e o Total Aptidão TA
- Se $Q_{h-1} < r < Q_h$, então cromossomo X_h é selecionado.

A aptidão de C é calculada pelo valor FO mostrada na expressão (4.11). TA recebe o valor correspondente a soma de todas aptidões C_h de cada solução da população.

$$TA = \sum_{h=1}^{TamP} C_h; \quad (4.11)$$

Conforme a expressão (4.12), PB corresponde a um peso que, é a razão entre o total das aptidões TA e aptidão de cada solução C_h .

$$PB = \frac{TA}{C(h)} \quad h = 1, 2, \dots, TamP. \quad (4.12)$$

O peso cumulativo é calculado pela Q_h de cada indivíduo da população. O valor acumulado atual é a soma dos PB anteriores e assim sucessivamente, até atingir o valor total das aptidões TA , conforme a expressão (4.13).

$$Q_h = \sum_{j=1}^h PB_j \quad h = 1, \dots, TA. \quad (4.13)$$

A Figura 21 ilustra um exemplo de roleta. A primeira coluna traz o índice do elemento da população que é uma solução candidata para ser sorteada. A coluna dois é representada pelo valor da aptidão de cada solução. A coluna três é preenchida com o peso, que é o cálculo descrito na Fórmula (4.12). Já na coluna quatro, é mostrado o valor do peso acumulado conforme a Fórmula (4.13). A última coluna traz os valores inteiros de um até o maior valor acumulado dos pesos. Na primeira linha, o peso 2, do indivíduo 1, é referente à razão do valor total da aptidão 1200 e o valor da aptidão do indivíduo 1. O valor 18, na última linha, e coluna 4, corresponde a soma dos pesos acumulados. Com isso, o indivíduo 1 tem chance 2 na roleta, o indivíduo 2 tem chance 5 na roleta e o indivíduo 3 tem chance 11 na roleta. Na hipótese do valor da roleta sorteado for 6 o indivíduo escolhido é o 2.

Figura 21: Exemplo de uma estrutura que armazena a roleta

Indivíduos	Aptidão	Peso	Peso Acumulado	Roleta
1	600	2	2	1 2
2	400	3	5	3 ... 7
3	200	6	11	8 .. 18
	1200	11	18	18

4.3.4.4 Realizar Cruzamentos

Foram escolhidos 3 operadores para implementação: o operador de cruzamento parcialmente mapeado (PMX), operador TCX e o operador OX. Conforme variáveis de decisão do OX, PMX e TCX o sistema pode executar os operadores em sequência, ou individualmente. Caso eles sejam executados em sequência o algoritmo controla para que fique sempre com a melhor solução entre as encontradas como resultado dos operadores.

1. PMX

A utilização do operador de cruzamento PMX segue a explicação já descrita na seção 3.3.6.2. A Figura 22, mostra como ficou o cruzamento PMX considerando os veículos

e depósitos, conforme abordado neste trabalho. Aqui, foi representada apenas a geração de um filho 1, porém o processo para a geração do filho 2 é igual.

Figura 22: PMX. Filho 1, Veículos e Depósitos

Pai	1	4	7	9	6	3	2	5	8
Veículo	1	2	2	2	2	2	3	3	4
Depósito	1	1	1	1	1	1	2	2	2
Mãe	9	3	5	7	2	8	6	4	1
Veículo	1	2	2	2	2	2	3	3	3
Depósito	1	1	1	1	1	1	2	2	2
Filho 1	1	4	9	7	2	8	6	5	3
Veículo	3	3	1	2	2	2	3	2	3
Depósito	2	2	1	1	1	1	2	1	2

2. TCX

A utilização do operador de cruzamento TCX segue a explicação já descrita na seção 3.3.6.3. A Figura 23, mostra como ficam o filho após o cruzamento TCX. Observa-se que nenhum cliente trocou de veículo na solução filho, pois permanecem com os mesmos veículos que estavam nas soluções mãe e pai. Ao final, se a solução necessitar de mais veículos, outro veículo poderá ser incluído. Na Figura 23, foi representada a geração de um filho, porém, o processo para a geração do outro filho é igual.

Figura 23: TCX. Filho com Clientes, Veículos e Depósitos

Mãe	9	7	5	6	2	8	4	3	1
Veículo	1	1	1	1	1	2	2	3	3
Depósito	1	1	1	1	1	2	2	3	3
Pai	8	2	3	7	5	1	9	4	6
Veículo	1	1	1	2	2	2	2	3	3
Depósito	1	1	1	2	2	2	2	3	3
Filho	7	5	2	3	8	4	9	1	6
Veículo	1	1	1	1	2	2	2	3	3
Depósito	1	1	1	1	2	2	2	3	3

3. OX

A utilização do operador de cruzamento OX segue a explicação já descrita na seção 3.3.6.1. Conforme a Figura 24, os movimentos ocorrem no nível de clientes fazendo com que estes troquem de veículos e até de depósito. Um procedimento verifica se as restrições continuam sendo respeitadas e realizando os ajustes necessários como a inclusão de veículo ou a realocação dos clientes nos mesmos veículos. Isso fará com que a solução se mantenha viável. Na Figura 24, foi representada apenas a geração de um filho, porém o processo para a geração do outro filho é igual.

Figura 24: OX. Filho com Clientes, Veículos e Depósitos

Pai	3	5	6	2	1	8	9	4	7
Veículo	1	1	2	2	2	2	3	3	3
Depósito	1	1	1	1	1	1	2	2	2
Mãe	9	6	3	1	4	7	8	2	5
Veículo	1	1	2	2	2	2	3	3	3
Depósito	1	1	1	1	1	1	2	2	2
Filho	9	3	6	2	1	8	4	7	5
Veículo	1	2	2	2	2	2	3	3	3
Depósito	1	1	1	1	1	1	2	2	2

4.3.4.5 Realizar as Mutações

Neste trabalho, foi utilizado o *Swap* como estratégia de mutação. Um cliente em um determinado nodo da rede é trocado por outro cliente pertencente ao mesmo depósito. Este procedimento recebe um cromossomo que representa os clientes, os veículos e o depósito. O processo do movimento de swap ocorre com a movimentação dos clientes. Dois genes diferentes são escolhidos aleatoriamente e assim são trocados de posição. Em seguida é realizada uma verificação que garante que a restrição de capacidade do veículo ainda é respeitada. Caso necessário, uma nova organização dos veículos é feita, e se preciso um novo veículo pode ser incluído, garantindo assim que a solução continue viável.

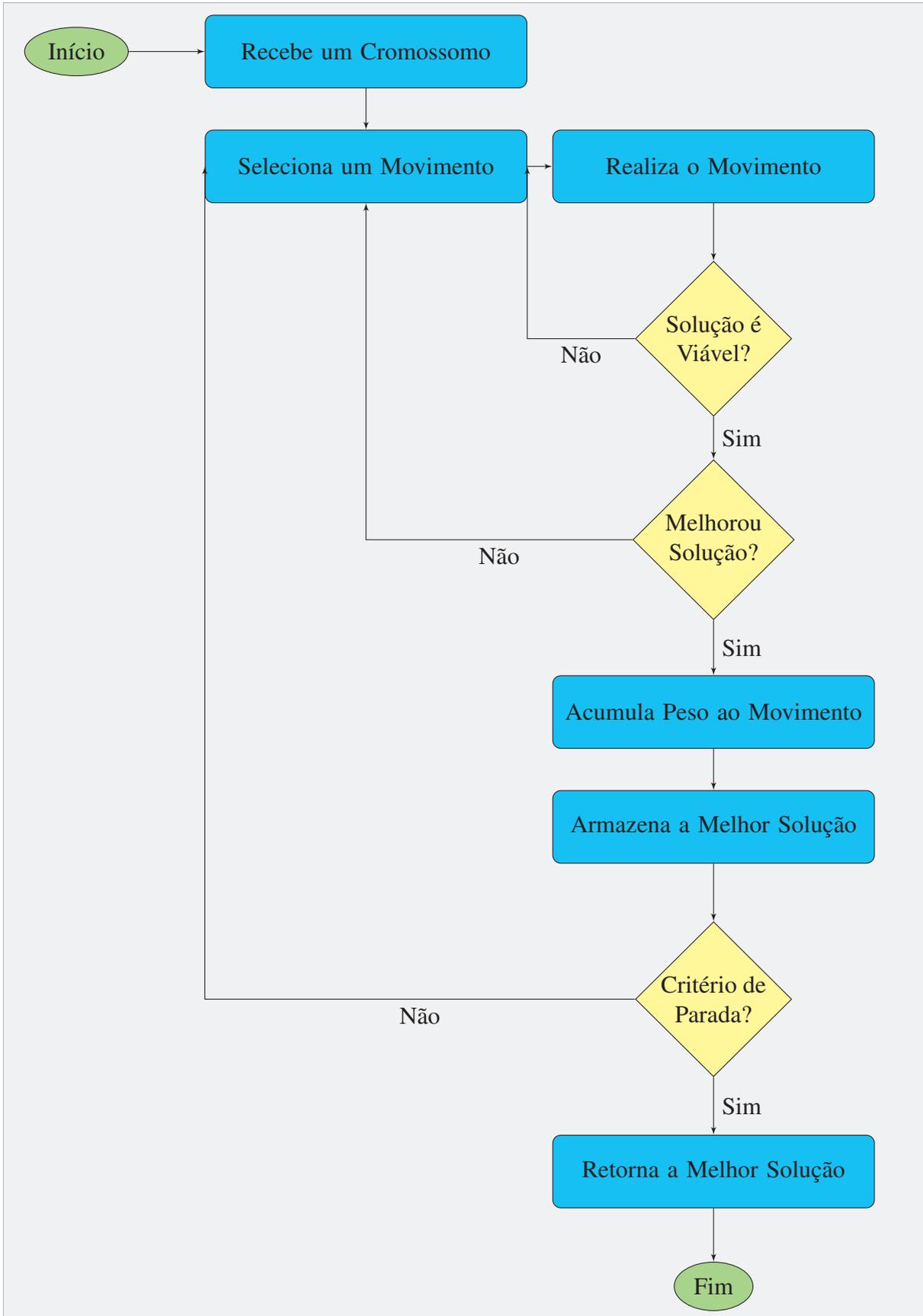
A Figura 25 mostra a representação do operador mutação Troca (*Swap*).

Figura 25: Mutação Troca

Clientes	3	5	6	2	1	8	9	4	7
Veículos	1	1	2	2	2	2	3	3	3
Depósitos	1	1	1	1	1	1	2	2	2
Clientes	3	5	8	2	1	6	9	4	7
Veículos	1	1	2	2	2	2	3	3	3
Depósitos	1	1	1	1	1	1	2	2	2

4.3.4.6 Realizar à Busca Local

O objetivo deste procedimento é realizar uma busca local, a fim de realizar uma intensificação no espaço de soluções. Inicialmente, realiza uma seleção para uma das técnicas de movimentos, conforme um peso dado durante as iterações. É avaliado se um dos movimentos propõe uma melhora, por isso, a cada iteração é incrementado um valor para o peso de cada movimento. Desta maneira, os movimentos que vão tendo melhora têm uma maior probabilidade de serem escolhidos a cada iteração.

Figura 26: Fluxograma demonstrando a sequência do ABL.

No ABL, uma estrutura armazena cada um dos movimentos e os seus respectivos pesos. Um parâmetro controla a quantidade de repetições que serão executadas. Para cada movimento escolhido, um procedimento é executado, onde é realizado um rearranjo no cromossomo. Então, é devolvido o resultado que corresponde ao valor da FO da solução nova proposta pelo movimento no rearranjo. Caso tenha melhorada a solução, o cromossomo e o valor da FO são armazenados como melhor solução dentro do ABL. O critério de parada é controlado por um parâmetro, atribuído no início da execução do algoritmo. Um procedimento verifica se a solução é viável quanto à restrição da capacidade dos veículos, caso não seja, é descartada. Ao final, a função retorna a melhor solução e o valor da FO correspondente. A melhor solução corresponde a uma sequência dos clientes os veículos e os depósitos. É mostrada a Figura 26, que corresponde ao fluxograma do ABL.

A seguir, é mostrado um exemplo de um dos movimentos escolhidos. Um procedimento, realiza o movimento, que remove ab e insere ba depois de c . Este procedimento recebe um cromossomo com clientes, veículos e depósitos. O objetivo principal é selecionar dois genes, aleatoriamente, a e b em posições sequenciais e inseri-los, de forma invertida, b e a depois de uma posição do gene c . O procedimento considera uma cadeia sequencial de genes de forma circular, ou seja, se a posição c , por exemplo, for a última, os dois genes sorteados serão inseridos na primeira e na segunda posição do cromossomo. A condição para que o procedimento funcione é de que tenha pelo menos 5 genes. Caso umas das posições destino, subsequentes à posição c , coincidam com posições de origem, o procedimento procura até encontrar posições factíveis.

Na Figura 27, os genes sorteados para serem retirados são 6 e 8 localizados nas posições 3 e 6 respectivamente. Os dois genes serão inseridos de forma invertida nas posições 1 e 2, que são as posições subsequentes da posição c , que é 9. Isso ocorreu porque a posição 9 é a última do cromossomo. Os outros movimentos têm comportamento semelhante a este diferenciando na forma de inversão dos genes sorteado para serem retirados, ou na quantidade de genes sorteados.

Figura 27: Movimento Remove Insere Depois

			A			B			C
Clientes	3	5	6	2	1	8	9	4	7
Veículos	1	1	2	2	2	2	3	3	3
Depósitos	1	1	1	1	1	1	2	2	2
Clientes	8	6	3	5	2	1	9	4	7
Veículos	2	2	1	1	2	2	3	3	3
Depósitos	1	1	1	1	1	1	2	2	2

Na Figura 28, é descrito o código fonte do ABL em linguagem de programação C. Neste código, é possível observar que é realizada uma seleção do movimento, e conforme o resultado da função de seleção, um movimento é escolhido. Os movimentos com maior probabilidade são aqueles que tiveram um resultado melhor nas iterações anteriores.

Figura 28: Código Fonte do ABL

```

1 #define PESO 10
2 int mat_movimentos[TAM_MOVI][PARAM_MOVI];
3 void busca_local_meme (int filho,mae){
4     for (int i=0; i < meme_movimentos; i++){
5         inicializaSolucao(filho,sol_aux);
6         /*Seleciona o movimento com a maior probabilidade de melhora.*/
7         ind_movimento=seleciona_movimento(mat_movimentos);
8
9         switch(ind_movimento) {
10            case 0:remove_a_insere_b (filho,cromossomo,
11                pop_cromossomo_modif);
12                break;
13            case 1: remove_ab_insere_ab_depois_c(filho,cromossomo,
14                cromossomo_modif);
15                break;
16            case 2:remove_ab_insere_ba_depois_c(filho,cromossomo,
17                cromossomo_modif);
18                break;
19            default : -1;
20        }
21    }
22    viavel=solucao_eh_viavel(ind_filho, qtd_veiculo,sol_aux,
23        cromossomo_modif);
24    result_fo=avalia_solucao(filho,sol_aux );
25    if ( viavel > 0 && result_fo < melhor_fo){
26        melhor_fo = result_fo;
27        copia_solucao(sol_melhor, sol_aux, filho);
28        /*Aumenta o peso do movimento que tem a melhora.*/
29        mat_movimentos[ind_movimento][1]+=PESO;
30    }
31    copia_solucao(solucao,sol_melhor,filho);
32 }

```

Conforme a Figura 28, no momento de seleção, uma rotina vai acumulando os pesos que serão utilizados para saber qual movimento tem a maior probabilidade de ser escolhido. A técnica de escolha é semelhante à roleta utilizada na seleção no AG. Logo após o procedimento realizar as mudanças no cromossomo, outro procedimento avalia se a nova solução é viável, ou seja, considera as restrições já estabelecidas como premissas, como por exemplo, a capacidade do veículo. Assim, é avaliado se a solução nova tem um resultado melhor do que a melhor armazenada até o momento dentro ABL. Caso a solução seja viável, e foi melhorada, ela é armazenada como melhor e o movimento recebe um incremento no peso. Ao final do ABL, a solução é copiada para a sequência do AG, melhorando ou não a solução de um dos indivíduos da população.

4.3.4.7 Avaliar População

Este procedimento recebe uma estrutura que representa a população. Cada indivíduo da população é uma solução. A solução traz a sequência dos clientes com os veículos que irão atendê-los e seus respectivos depósitos. Para cada solução também é armazenado o valor de aptidão que é o valor da função objetivo. Durante a execução dos operadores do algoritmo genético caso o filho tenha uma aptidão melhor que o pai ou que a mãe a solução filho substitui a solução pai na população. Desta forma as boas soluções podem estar no início, no fim ou no meio da população. Este procedimento tem o objetivo de classificar a população pelo valor da aptidão de forma que as melhores soluções estejam no início.

4.3.4.8 Guardar Soluções Elite

Este procedimento, a partir da classificação da Matriz da população, seleciona as E melhores soluções elite. A ideia é deixar na população um número de soluções boas que possam contribuir para a melhoria da população através das futuras gerações. Na primeira iteração do algoritmo genético a solução elite é formada com a repetição da solução inicial e mais soluções aleatórias com a devida ordenação. A partir da segunda iteração a população elite baseia-se na população atual classificada.

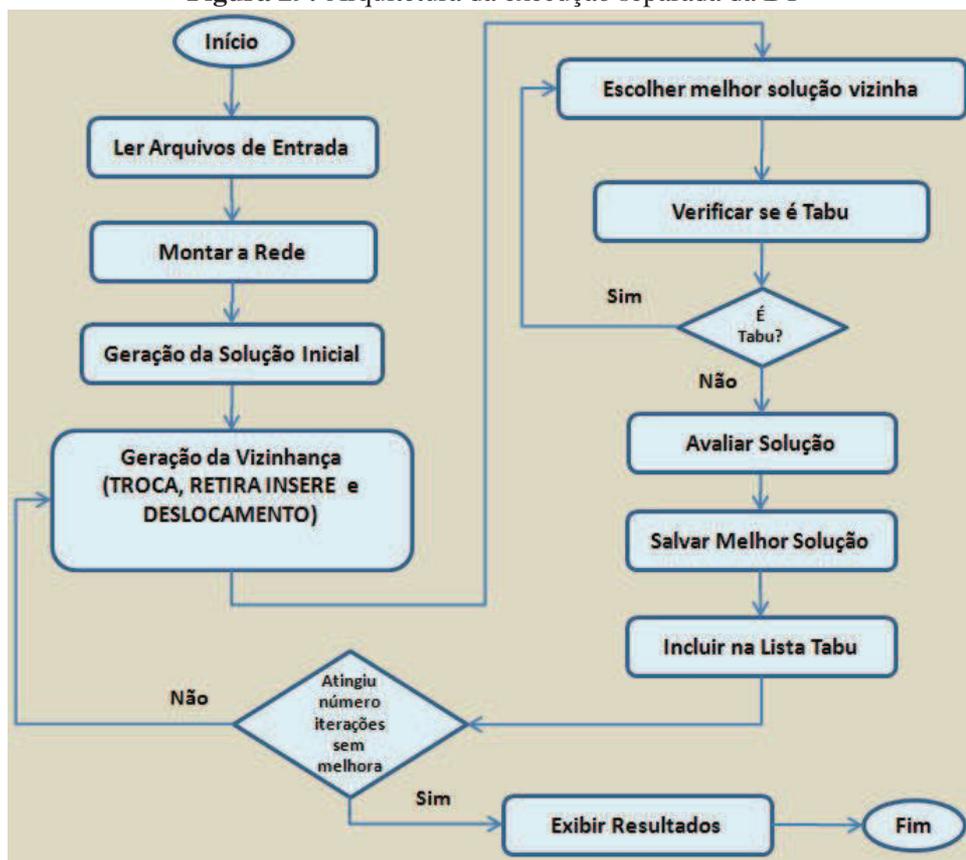
4.3.4.9 Geração da População com Soluções Elite

O procedimento controla para que a população elite represente sempre as E melhores soluções encontradas na população N . O restante da população $N - E$ é construída de forma em que a sequência dos clientes seja gerada aleatoriamente.

4.3.5 Execução somente do BT

O procedimento recebe a solução do AG, caso este seja executado; senão recebe a solução inicial. A melhor solução encontrada é armazenada como uma solução elite. O algoritmo é executado enquanto o número de iterações sem melhora não for ultrapassado. Este procedimento devolve a melhor solução encontrada. A melhor solução é armazenada entre a solução inicial, a solução do AG ou a solução da BT, para cada iteração do AH. A melhor solução encontrada no algoritmo, que é a primeira posição das soluções armazenadas na estrutura que corresponde à população, coincide com a melhor solução do algoritmo híbrido. Conforme a Figura 29, os procedimentos utilizados neste módulo são: Gerar Soluções de Vizinhança, Escolher Melhor Vizinho, Verificar se solução tem Movimento Tabu, Avaliar Critério de Aspiração, Incluir na LT, Avaliar Solução.

Figura 29: Arquitetura da execução separada da BT



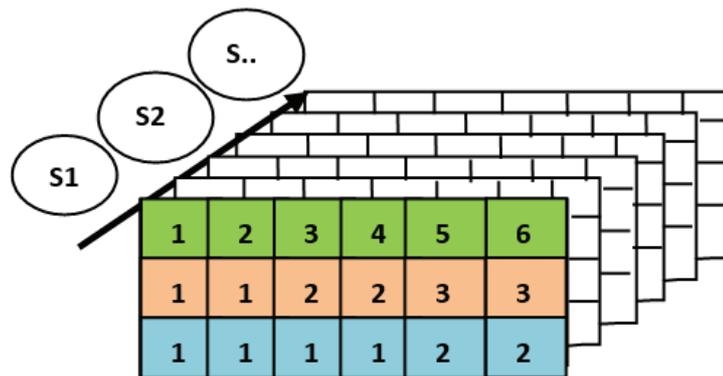
O procedimento recebe a melhor solução encontrada no algoritmo genético, caso ele tenha sido executado anteriormente. Caso não haja execução do algoritmo genético antes, uma solução inicial é utilizada para começar o procedimento Busca Tabu. Na Figura 29, está a representação do procedimento BT caso seja executado separadamente. O procedimento vai ser executado um número de vezes que seja menor do que o parâmetro "Max Tabu", ou seja, uma quantidade máxima de iterações sem melhora do valor da função objetivo. A cada iteração, uma

solução é enviada à um procedimento, para gerar uma vizinhança. Com isso, através de um movimento de perturbação em soluções vizinhas, é escolhida uma solução que poderá melhorar o valor da FO, em relação as anteriores. A cada iteração a solução é armazenada em uma lista de movimentos proibidos denominada LT. O procedimento verifica, se, na solução da iteração corrente houve uma melhora no valor da função objetivo. Caso ocorra a melhora, a solução e o valor da FO são armazenados.

4.3.5.1 Geração de Vizinhança

Procedimento responsável por gerar uma pesquisa no espaço de busca para encontrar soluções que deverão ser avaliadas e escolhidas, caso possuïrem uma qualidade boa. Cada vizinho gerado corresponde a uma solução, ou seja, uma sequênciã dos clientes, os depósitos e os veículos.

Figura 30: Vizinhança



Os movimentos são realizados com o rearranjando dos clientes, que corresponde a primeira linha na Figura 30. Como os clientes poderão ser trocados de posição no novo arranjo, estes poderão fazer parte de um novo veículo (inter-rota) e/ou um novo depósito (intra-rota). A vizinhança é estruturada da seguinte forma:

- a) Troca intra-rota com o mesmo depósito;
- b) Troca inter-rota com o mesmo depósito;
- c) Troca inter-rota com troca do depósito;
- d) Retira e Insere intra-rota com o mesmo depósito;
- e) Retira e Insere inter-rota com o mesmo depósito;
- f) Retira e Insere inter-rota com troca do depósito.
- g) Deslocamento intra-rota com o mesmo depósito;

- h) Deslocamento inter-rota com o mesmo depósito;
- i) Deslocamento inter-rota com troca do depósito.

1. Movimento Troca

O movimento troca, conforme mostra a Figura 31, recebe uma solução representada por um cromossomo com clientes, veículos e depósitos para que seja realizado um movimento chamado *swap*. A solução é armazenada em uma matriz e, para cada solução, um item, que corresponde a um cliente é sorteado aleatoriamente. Logo após, uma outra posição, diferente da primeira é escolhida também aleatoriamente. O valor da primeira posição é inserido na posição da segunda posição sorteada e o valor da segunda posição é inserido na posição da primeira. Com as trocas, um veículo poderá receber mais clientes que tinha antes a movimentação, desta forma, a capacidade poderá ser ultrapassada. Com isso, é realizada uma verificação que garanta que a restrição de capacidade do veículo seja respeitada. Então uma nova organização dos veículos é feita, e se preciso um novo veículo pode ser designado, garantindo que a solução permaneça viável. Na Figura 31, segue um exemplo de um movimento de troca, com algumas soluções possíveis. O movimento considera apenas os clientes, ou seja, para fins de representação, não são mostrados os veículos e os depósitos. O número total de soluções geradas com este movimento é representado pela expressão (4.14).

$$VT = \frac{N * (N - 1)}{2} \quad (4.14)$$

Figura 31: Lista de candidatos para o Movimento Troca

Solução	1	2	3	4
1)	3	2	1	4
2)	1	4	3	2
3)	1	3	2	4

2. Movimento Retira e Insere

O movimento Retira e Insere recebe uma solução representada por uma sequência de clientes, veículos e depósitos para que seja realizado um movimento. O procedimento desloca cada um dos genes do cromossomo para uma posição posterior. O movimento realizado é circular, ou seja, o último elemento da solução ocupará a primeira posição. O número total de soluções geradas com este movimento é representado pela expressão (4.15). Após a geração do movimento é realizada uma verificação que garanta que a restrição de capacidade do veículo seja respeitada. Se necessário, uma nova organização dos veículos é feita, e se preciso um novo veículo pode ser designado, garantindo que a solução ainda permaneça viável. Na Figura 32, é mostrado um exemplo de algumas soluções geradas pelo movimento Retira e Insere.

$$VMI = \frac{N * (N - 1)}{2} \quad (4.15)$$

Figura 32: Lista de candidatos para o Movimento Retira e Insere

Solução	1	2	3	4
1)	1	3	4	2
2)	1	3	2	4
3)	1	2	4	3

3. Movimento Deslocamento

O movimento deslocamento recebe uma solução representada por um cromossomo com clientes, veículos e depósitos, para que seja realizado um movimento chamado *shift*. O procedimento desloca cada elemento da solução para uma posição posterior. O movimento circular é considerado, ou seja, o último elemento do cromossomo ocupará a primeira posição. Em seguida, é realizada uma verificação que garanta que a restrição de capacidade do veículo seja respeitada. Se necessário, uma nova organização dos veículos é feita, e se preciso um novo veículo pode ser designado, garantindo que a solução ainda permaneça viável. Na Figura 33, é mostrado um exemplo do movimento Movimento Deslocamento. O movimento de deslocamento gera M possibilidades.

Figura 33: Lista de candidatos para o Movimento Deslocamento

Solução	1	2	3	4
1	4	1	2	3
2	3	4	1	2
3	2	3	4	1

4.3.5.2 Escolher melhor solução vizinha

Este procedimento recebe a melhor solução corrente que sempre é armazenada no algoritmo Busca Tabu, a cada interação. A solução corrente é utilizada para que seja gerada uma vizinhança. A vizinhança representa um conjunto de soluções que são concebidas a partir de movimentos de rearranjos com base na solução corrente. São utilizados parâmetros de controle que determinam qual estratégia de movimento é realizada na geração da vizinhança. As estratégias são troca, Retira e Insere e deslocamento. Com a vizinhança gerada, é feita uma avaliação de cada uma das soluções para que seja escolhida aquela que melhora o valor da função objetivo, em relação a melhor solução. No procedimento, também é verificado se a solução possui um movimento incluído na LT, ou seja, é um movimento proibido. As estratégias podem ser executadas em sequência, caso isso ocorra, o procedimento retorna à melhor solução dentre as estratégias escolhidas. A solução é armazenada se tiver um resultado melhor dentre as soluções

candidatas, se não infringir nenhuma restrição e se não for um movimento tabu com status de proibido.

4.3.5.3 Verificar se é Tabu

Neste procedimento é feito um acesso à LT onde é realizada a verificação se a solução analisada tem movimentos inversos em relação as solução armazenadas, então é considerado um movimento tabu. Se este movimento inverso for percebido, o status deste movimento é considerado proibido.

4.3.5.4 Avaliar Solução

A FO fornece um valor que corresponde as distancias entre clientes (i, j) e custo fixo do veículo. Deve-se observar que, para cada rota a soma dos pedidos dos clientes não ultrapasse a capacidade do veículo. O cálculo do valor FO é representado pela distância total da rota de cada veículo, que considera as distâncias entre os clientes, a distância entre o depósito e o primeiro cliente e mais a distância do último cliente e o depósito. O valor da FO considera a soma total das distâncias e mais a soma total dos custos fixo para cada veículo utilizado na solução.

4.3.5.5 Avaliar Critério de Aspiração

Neste trabalho, o critério de aspiração está relacionado à avaliação da função objetivo (FO) correspondente à melhor solução conhecida até o momento conforme descrito na subsecção 4.3.5.4.

4.3.5.6 Incluir na Lista Tabu

A maneira como é configurada a LT, é armazenada a solução, conforme está representado na Figura 34. São armazenados a cada iteração os n soluções que representa a melhor escolha obtida na exploração da vizinhança a cada iteração, mesmo que esta não seja a melhor global de todo o algoritmo. As soluções são incluídas até esgotar o tamanho da LT, e a partir daí, por ser circular, a solução é incluída no fim da fila, ou seja, a primeira a ser retirada foi a primeira a ser incluída, e assim sucessivamente.

Figura 34: Inclusão na LT

Origem	2	3	4	5	1	6
Destino	4	3	2	5	1	6

A Figura 34 mostra a troca de posição de um cliente. No primeiro momento, há um movimento da solução origem para a solução destino. Neste movimento ocorrem as trocas da

primeira posição e da terceira posição, ou seja, os valores 2 e 4 ficaram invertidos. Na solução origem, o valor 2 estava na posição 1 e passou para a posição 3 na solução destino. Na solução origem, o valor 4 estava na posição 3 e passou para a posição 1 na solução destino. Então a solução origem é armazenada como solução tabu. Com isso, caso a solução destino realize um movimento inverso trocando o valor 2 que está na posição 3 para a posição 1 e o valor 4 que está na posição 1 para a posição 3, então gerará uma solução igual à solução origem que é considerada um movimento proibido.

A Figura 35 representa a solução origem e a Figura 36 representa a solução destino. O movimento será considerado proibido caso, em uma outra iteração, a solução mostrada na Figura 36 gere um vizinho igual a solução exibida na Figura 35.

A Figura 36 mostra a solução destino com os veículos que farão o atendimento dos clientes e os depósitos de cada veículo. É possível perceber que o cliente 2 trocou de posição da solução origem para a solução destino, trocando também de veículo e mantendo o depósito, já que tanto o veículo 1 quanto o veículo 2, pertencem ao mesmo depósito. O mesmo comportamento ocorre com o o cliente 4.

Figura 35: Solução origem

Cientes	2	3	4	5	1	6
Veículos	1	1	2	2	3	3
Depósitos	1	1	1	1	2	2

Figura 36: Solução destino

Cientes	4	3	2	5	1	6
Veículos	1	1	2	2	3	3
Depósitos	1	1	1	1	2	2

Neste capítulo foram apresentados os principais procedimentos que tiveram o objetivo de executar o AH a partir de arquivos já conhecidos. No capítulo 5.1 são abordadas as estratégias de validação do modelo. Já na seção 6.1 são exibidos os conjuntos de dados utilizados nos algoritmos e na seção 6.2 são mostrados os resultados obtidos a partir dos critérios de otimização, contemplados nos algoritmos apresentados neste trabalho.

5 PLANEJAMENTO DOS EXPERIMENTOS

Neste capítulo, são apresentadas as estratégias para validar o modelo apresentado. São descritos os diferentes tipos de experimentos realizados, bem como os parâmetros que são responsáveis pela calibração do AG, da BT e do AH.

5.1 Ajuste de Parâmetros

Os parâmetros na BT, onde foram realizados os ajustes, são:

- a) Tamanho da LT (*tamTabu*): é o número de soluções consideradas Tabu, armazenadas em uma lista.
- b) Número de Iterações sem Melhora (*nbMax*): é o número de iterações sem que haja melhora do valor da FO.

Diferentes experimentos foram realizados para calibrar os parâmetros dos algoritmos, utilizando os programas teste de Cordeau, Gendreau e Laporte (1997), com os arquivos de 50, 75, 100, 160, 240 e 360 clientes.

Alguns experimentos foram executados, levando em conta o tamanho da LT e o número de execuções do algoritmo BT, conforme já foi apresentado no trabalho de Simas e Gómez (2007).

1. **Diferentes tamanhos de instâncias:** Os algoritmos apresentados neste trabalho utilizaram as instâncias 50, 75, 100, 160, 240 e 360.
2. **Valores para o nbMax na BT:** O algoritmo BT foi executado para diferentes valores do *nbMax* e são eles: 100, 250, 500, 750, 1000, 1250, 1500.
3. **Valores para o tamTabu:** O algoritmo BT foi executado, para diferentes valores da LT que são eles: 10, 25, 50, 75, 100, 150, 200.

Diferentes Estratégias de Vizinhança foram utilizadas para o algoritmo BT, levando em conta que neste trabalho, foi considerado o Problema de Roteamento de Veículos com Múltiplos Depósitos e Frota Heterogênea (PRVMDFH).

Os parâmetros *tamPop*, *numGera*, *taxaCruza* e *taxaMuta* foram utilizados para ajuste no AG, como descrito a seguir:

- Tamanho da população (*tamPop*): quantidade de indivíduos (cromossomos) utilizada a cada geração;
- Número de gerações (*numGera*): quantidade de vezes em que uma população é modificada para ser avaliada;

- Taxa de cruzamento (taxaCruza): taxa utilizada para realizar o operador de cruzamento.
- Taxa de mutação (taxaMuta): taxa utilizada para realizar o operador de mutação.

Diferentes experimentos foram realizados para calibrar os parâmetros do algoritmo, entre eles os programas teste de Cordeau, Gendreau e Laporte (1997), onde foram utilizados os arquivos com 50, 75, 100, 160, 240 e 360 clientes.

- a) **Varição da Taxa de Cruzamento:** foram testadas as seguintes taxas de cruzamento, entre elas: 0.80, 0.85 e 0.90,
- b) **Varição da Taxa de Mutação:** foram testadas as seguintes taxas de mutação: 0.05, 0.03, 0.02 e 0.01.
- c) **Varição do Número de Gerações:** foram testados os seguintes valores para o número de gerações: 100, 250, 500, 750, 1000 e 1500.
- d) **Diferentes operadores de Cruzamento:** foram utilizados os operadores de cruzamento PMX, TCX e OX
- e) **Operadores de Mutação:** foi utilizado o operador de mutação Troca.

6 RESULTADOS

O objetivo nesta etapa, através da análise de um conjunto de dados e com a comparação a outros modelos, foi verificar se a dinâmica do modelo apresentado neste trabalho está adequada para gerar bons resultados para o problema do PRVC.

6.1 Conjunto de Dados

O primeiro conjunto de dados do tradicional trabalho de Desrochers, Desrosiers e Solomon (1992), ou seja, aborda o PRVC e são eles: C101, R101 e R105 para instâncias de 25 clientes e C101, R101 e R105 para instâncias de 50 clientes. O objetivo da escolha destes arquivos era utilizar instâncias não muito grandes para validar o algoritmo com o PRVC sem considerar múltiplos depósitos e sem considerar frota heterogênea. Nestes arquivos, a primeira linha contém uma codificação com o nome do problema, o número de veículos e a capacidade dos veículos. Além disso, são utilizadas informações pertinentes a cada um dos clientes, como está descrito a seguir.

- o número do cliente;
- a coordenada x;
- a coordenada y;
- a demanda do cliente;

Os arquivos utilizados foram C101, R101 e R105 para 25 clientes e a capacidade do veículo de 200, e os arquivos c101, r101 e r105 para 50 clientes, também com a capacidade do veículo de 200.

O segundo conjunto de dados utilizado é referente ao trabalho do Cordeau, Gendreau e Laporte (1997) e são eles: p02, p03, p04, p15, p19 e p23. O objetivo da escolha é escolher instâncias pequenas, médias e grandes. Além disso, este conjunto de arquivos trata diretamente o PRV com múltiplo depósito. O conjunto de arquivos auxiliou na validação porque também é fornecida a sequência da melhor solução encontrada pelo autor. Nestes arquivos, no cabeçalho, são descritas as informações como o tipo de PRV, número de veículos, número de clientes e o número de depósitos. Na sequência do arquivo, estão os dados referente ao cliente, onde neste trabalho, foram utilizadas as seguintes informações:

- o número do cliente;
- a coordenada x;
- a coordenada y;

- a demanda do cliente;

O arquivo p02 corresponde a 50 clientes, com a capacidade do veículo de 160. Já o arquivo p03 traz 75 clientes, com a capacidade de 140. O arquivo p04 possui 100 clientes, tendo a capacidade igual a 160. O arquivo p15 traz 160 clientes e a capacidade igual a 160. O arquivo p19 é composto por 240 clientes e a capacidade de 60. Por fim, o arquivo p23 possui 360 clientes com a capacidade do veículo de 60.

Com estes arquivos de programas teste, foi possível adequar o algoritmo e torná-lo capaz de achar boas soluções para o problema proposto. Na seção 6.2, foram exibidos os resultados obtidos com a execução do AH.

6.2 Resultados Obtidos

Neste capítulo são apresentados os resultados da execução do AH, com base nas estratégias do modelo proposto MPRVDH. No primeiro momento são mostrados os dados referentes à execução do AG, onde foram utilizados como referência, os arquivos dos programas teste de Cordeau, Gendreau e Laporte (1997). Em seguida é exibida a análise referente ao número de gerações para a instância de 50 clientes. E por fim, ainda para o AG, é mostrado o comportamento do Algoritmo de Busca Local para a instância de 75 clientes. No segundo momento, os resultados são avaliados em função da execução do algoritmo BT, considerados múltiplos depósitos um tipo de veículo. Inicialmente é analisado o comportamento dos movimentos utilizados para geração da vizinhança, Troca e o Retira e Insere. Também é analisada a variação do tamanho da LT e a variação do número de iterações sem melhora do valor da FO. No terceiro momento, são mostrados os dados referentes à execução do AH.

São exibidos os resultados para 50 execuções do algoritmo para cada problema teste. É exibido o melhor resultado de cada instância, a média, o desvio padrão, o número de depósitos e o número de veículos que coincide com as rotas. Por fim, foi realizada uma simulação para múltiplos depósitos e para três tipos diferentes de veículo, com base nos problemas testes apresentado por Cordeau, Gendreau e Laporte (1997). Os algoritmos foram executados em uma máquina HP Z230 Workstation com um processador Intel(R) Xeon(R) CPU E3-1241 v3 @ 3.50GHz e memória instalada (RAM) 16,0 GB e com o sistema operacional Windows 10 Enterprise x64.

6.2.1 AG com Múltiplos Depósitos e Um Tipo de Veículo

Aqui são mostrados os experimentos com a execução do AG considerando mais que um depósito e um tipo de veículo, conforme os dados dos arquivos utilizados. A execução é baseada no conjunto de dados apresentado por Cordeau, Gendreau e Laporte (1997), que trata o PRV conforme é exibido na Tabela 1. Para comparar ao trabalho de Cordeau, Gendreau e Laporte (1997), foi considerado um tipo de veículo, de acordo com os problemas testes propostos pelo

autor. Os valores de entrada como o número de depósitos e quantidade de clientes, a capacidade dos veículos, coordenadas e pedidos dos clientes, foram os mesmos. Os problemas testes escolhidos foram p02, p03, p04, p15, p19 e p23. Os valores obtidos pelo AG aproximaram-se dos valores obtidos pelo trabalho de Cordeau, Gendreau e Laporte (1997), sendo a quantidade de veículos utilizada a mesma. Os valores utilizados nos parâmetros são: numGeração = 1000, tamPopulacao = 300, taxaCruzamento = 0.9 e taxaMutação = 0.02.

Tabela 1: Dados do AG comparado a Cordeau, Gendreau e Laporte (1997)

		Cordeau	AG	% Gap
Prob. Teste	Veículos	FO	FO	
p02	5	473,87	479,96	-1,2851
p03	11	641,20	653,75	-1,9572
p04	15	1006,66	1034,51	-2,7665
p15	16	2534,13	2594,72	-2,3909
p19	24	3827,06	3867,06	-1,0451
p23	32	6139,73	6205,98	-1,0790
FO = Distância Total				

Conforme a Tabela 1, o AG alcançou o mesmo número de rotas e valores da FO próximos ao trabalho de Cordeau, Gendreau e Laporte (1997). Nos resultados descritos na Tabela 2, pode-se observar o comportamento do número total de gerações no AG em relação a melhora da função objetivo. Os valores observados foram 100, 250, 500, 750, 1000 e 1500 gerações. Para isso, foram utilizados os valores para as taxas de cruzamento de 0.9, taxa de mutação de 0.02 e taxa de execução do Algoritmo de Busca Local de 0.02. Conforme os resultados mostrados a seguir, nota-se que, a partir 500 gerações quanto maior o número de gerações, melhor o resultado obtido em relação a FO.

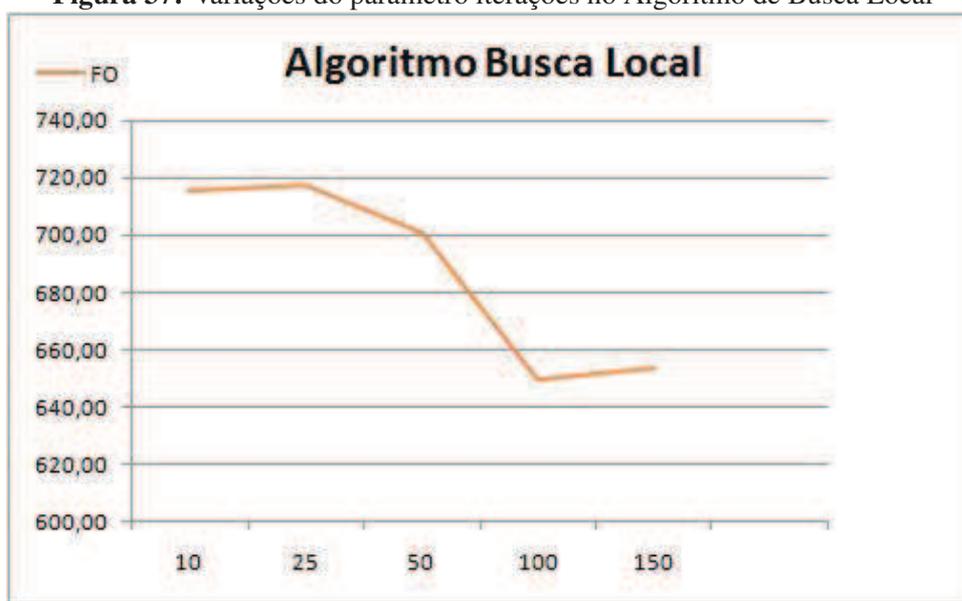
Tabela 2: Variações do parâmetro Número de Gerações

Cordeau p02 para 50 clientes						
Número de Gerações	100	250	500	750	1000	1500
Distância (FO)	494,19	495,35	495,35	498,54	490,62	479,96

Na Tabela 3 e Figura 37, é mostrada a variação do número de iterações da busca local no ABL. Desta forma, foi utilizado um parâmetro buscaLocal com os valores de 10, 25, 50 100, 150. Foi utilizado o arquivo de problema teste p03, referente ao trabalho de Cordeau, Gendreau e Laporte (1997). Para isso, foram utilizados os valores para as taxas de cruzamento 0.9, taxa de mutação de 0.02 e taxa para execução do Algoritmo de Busca Local de 0.02. De acordo com a Tabela 3, entre os valores 10 e 25 do parâmetro houve uma piora porém entre os valores 25 e 50 e os valores 50 e 100, a melhora ocorreu. O melhor resultado obtido neste experimento coincide com o melhor resultado encontrado na Tabela 1, pois, naquele experimento o ABL foi também executado no AG.

Tabela 3: Variações do parâmetro iterações no Algoritmo de Busca Local

Cordeau p03 para 75 clientes					
Iterações	10	25	50	100	150
FO	715,58	717,35	700,90	649,44	653,75

Figura 37: Variações do parâmetro iterações no Algoritmo de Busca Local

6.2.2 BT com Múltiplos Depósitos e Um Tipo de Veículo

Aqui, os resultados foram obtidos com base na execução do algoritmo BT, com a premissa de ser possível mais de um depósito e com um tipo de veículo, conforme dados dos arquivos utilizados. Os experimentos tratam primeiro de uma comparação entre os movimentos na geração da vizinhança. São considerados movimentos na mesma rota, em rotas diferentes ou até em rotas diferentes e depósitos diferentes. Na Tabela 4, são observados os resultados, considerando dois tipos de movimentos utilizados no algoritmo BT: o movimento Troca e o movimento *Retira Insere*. A ideia é realizar um paralelo entre a utilização de uma estratégia em relação a outra. O problema teste utilizado foi p02, com a instância de 50 clientes, conforme o trabalho de Cordeau, Gendreau e Laporte (1997). Os valores utilizados nos parâmetros foram: nbMax=1000, tamList=150. Neste experimento, pode-se observar que, a utilização do movimento Troca obteve resultados melhores do que o movimento Retira e Insere, para geração da vizinhança. Para esse experimento, o algoritmo executou os movimentos separadamente, e os parâmetros restantes permaneceram os mesmos. É considerado aqui apenas os valores da distância total percorrida para a avaliação da FO. Para a análise, foi mostrado apenas o valor da FO, porém, para todas os resultados foram obtidas 5 rotas e 4 depósitos, conforme os arquivos de programas teste publicado pelo autor. Aqui, observou-se a estratégia de intensificação onde o algoritmo BT concentra-se mais intensamente na exploração de regiões anteriormente encontradas.

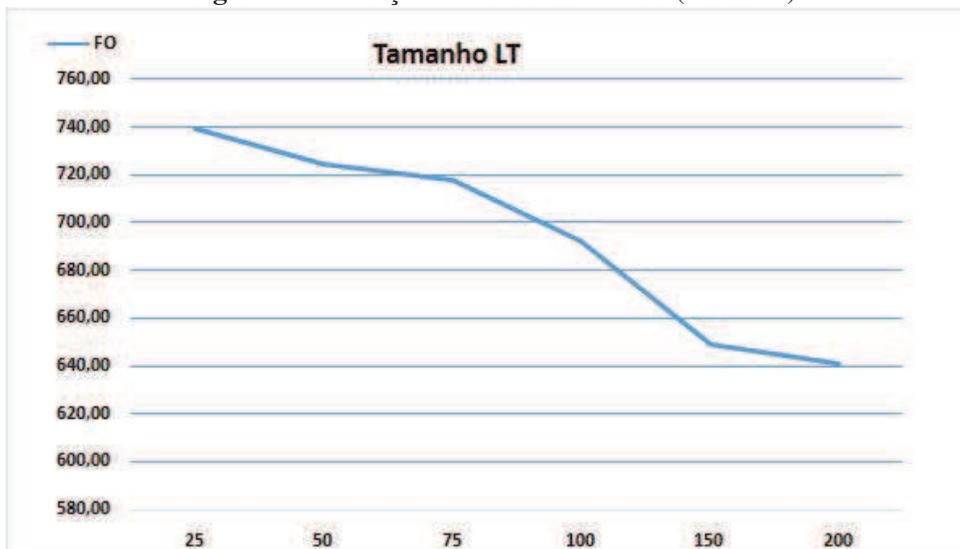
Tabela 4: Variações do movimento de vizinhança Busca Tabu

Cordeau p02 50 clientes						
	Troca			Retira Insere		
Depósito	mesmo	mesmo	outro	mesmo	mesmo	outro
Movi	intra	inter	inter	intra	inter	inter
Distância	473,87	479,96	487,29	489,51	492,79	494,19

Observa-se, na Tabela 5 e Figura 38, o comportamento do tamanho da LT. Os valores avaliados foram 25, 50, 75, 100, 150 e 200. O parâmetro nbMax mantém-se o mesmo para cada diferente tipo de tamanho da lista, neste caso foi utilizado o valor 100. A ideia foi observar o comportamento do valor da FO, que neste experimento, leva em consideração apenas a distância total percorrida pelos veículos entre o depósito e o cliente e entre o cliente e outro cliente. Observa-se que, conforme foi aumentando o tamanho da LT houve uma melhora em relação ao valor da FO, ou seja, quanto maior o valor da LT melhor o resultado. O algoritmo, independente de qual estratégia escolhida manteve apenas uma LT. Pode observar-se a estratégia da diversificação, que, segundo Crino et al. (2004), é um processo que necessita explorar diferentes regiões até o momento pesquisadas. A diversificação afasta as atuais áreas de pesquisa local, onde movimentos tendem a ser sem melhora. Aqui na BT, as soluções sem melhoras são iterativamente armazenadas na LT, com isto, a estratégia de diversificação força a busca em outras regiões ainda não exploradas.

Tabela 5: Variações do Tamanho da LT

Cordeau p03 para 75 clientes						
Tam.LT	25	50	75	100	150	200
Distância Total	739,67	725,05	717,77	692,16	649,58	641,20

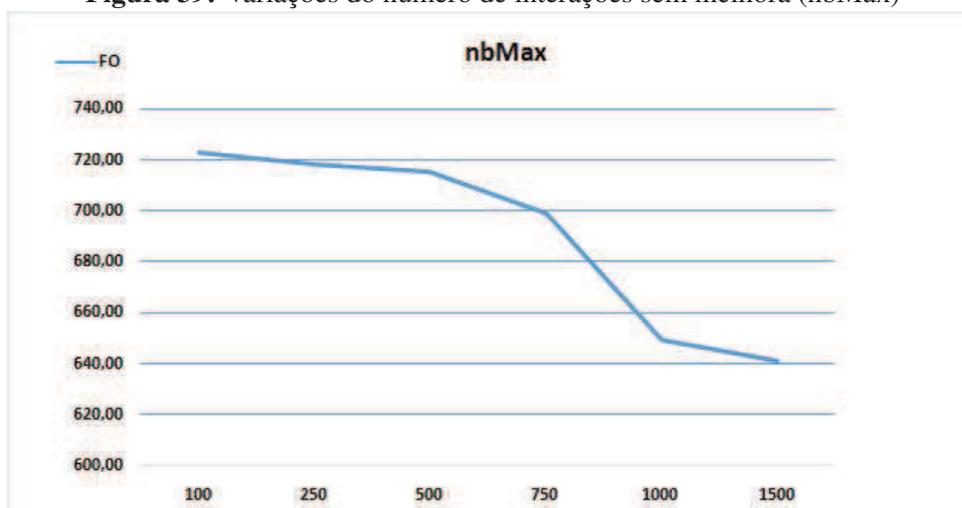
Figura 38: Variações do Tamanho da LT (tamLista)

Na Tabela 6 e Figura 39, é mostrado o comportamento do nbMax, que é o número de iterações sem que haja melhora na função objetivo. A ideia foi observar a distância total percorrida pela solução na execução do algoritmo com a variação do nbMax. O parâmetro Tamanho da LT permaneceu o mesmo, para cada diferente valor do nbMax, neste caso foi utilizado o valor = 100. Neste experimento, não estão exibidos os valores correspondentes aos veículos e aos depósitos, desta forma somente o valor da FO foi considerado para observação. É nítido que, a partir do valor 100 até o valor 1500 para o nbMax há uma melhora do valor da FO. Em cada aumento do valor do nbMax, houve a diminuição da FO.

Tabela 6: Variações do número de interações sem melhora (nbMax)

Cordeau p03 para 75 clientes						
nbMax	100	250	500	750	1000	1500
Distância Total	723,10	718,40	715,64	699,10	649,58	641,20

Figura 39: Variações do número de interações sem melhora (nbMax)



6.2.3 AH Depósitos e Um ou Três Tipos de Veículos

Aqui, serão mostrados os resultados referentes a execução do AH que consiste em unir as execuções do AG e do BT ao mesmo tempo. No primeiro momento o algoritmo foi executado considerando apenas um depósito e um tipo de veículo, para os arquivos de programa teste de Desrochers, Desrosiers e Solomon (1992). No segundo momento são mostrados os dados utilizando múltiplos depósitos e com um tipo de veículo e no terceiro momento múltiplos depósitos com três tipos de veículo.

6.2.3.1 Com um Depósito e um Tipo de Veículo

Nesta seção são mostrados os resultados a partir da execução do AH, considerando apenas um depósito e um tipo de veículo. A Tabela 7 traz o conjunto de dados disponibilizados por Desrochers, Desrosiers e Solomon (1992). Neste experimento, foram utilizados os parâmetros: BT nbMax = 1000 e tamList = 150, no AG taxa de cruzamento = 0.9, taxa de mutação = 0.02 e taxa para execução do ABL = 0.02. Para o AH, o parâmetro utilizado foi maxHíbrido = 100. Este foi o primeiro experimento a ser realizado onde o objetivo principal era executar o algoritmo para a versão mais tradicional do PRVC, utilizando apenas um depósito e um tipo de veículo. O AH comportou-se de forma satisfatória, alcançando resultados semelhantes ao trabalho de Desrochers, Desrosiers e Solomon (1992).

Tabela 7: Dados Algoritmo Híbrido comparado a Desrochers, Desrosiers e Solomon (1992)

Prob. Teste	Clientes	Solomon		AH		Melhora %
		Veículos	FO	Veículos	FO	
C101	25	3	191,3	3	191,3	0
R101	25	8	617,1	8	617,1	0
R105	25	6	530,5	4	528,3	0,4147
C101	50	5	362,4	5	362,4	0
R101	50	12	1047,0	13	1036,4	1,0124
R105	50	9	906,6	9	897,3	1,0258
FO = Distância Total						

6.2.3.2 Com Múltiplos Depósitos e um Tipo de Veículo

Nesta seção, são mostrados os resultados a partir da execução do AH, considerando mais de um depósito e apenas um tipo de Veículo. Na Tabela 8, são mostrados resultados obtidos a partir da execução do AH para os problemas p02, p03, p04, p15, p19 e p23. Neste experimento o valor da FO é a soma total das distâncias entre os clientes e os depósitos, não considerando o custo fixo dos veículos. Foram utilizados os seguintes parâmetros: na BT nbMax = 1000 e tamList = 150, no AG taxa de cruzamento = 0.9, taxa de mutação = 0.01 e taxa para execução do ABL = 0.02. Para o AH, o parâmetro utilizado foi maxHíbrido = 100. Foi feita uma comparação do valor da FO do AH com o trabalho de Cordeau, Gendreau e Laporte (1997) e descrito o percentual de melhora da FO. Pode-se observar que nas instâncias p02 e p03 o resultado foi o mesmo, nas instâncias p15 e p23 houve uma melhora e na p19 o resultado piorou.

Na Tabela 9, são mostrados os melhores resultados para os conjuntos de Cordeau, Gendreau e Laporte (1997), bem como dados estatísticos como média e o desvio padrão. Os experimentos foram realizados 50 vezes para cada um dos problemas teste. Os arquivos de problemas testes escolhidos foram p02, p03, p04, p15, p19 e p23. O valor utilizado para a capacidade corresponde a um tipo de veículo apenas, conforme proposto pelo autor. Foram utilizados os seguintes

Tabela 8: Dados do AH comparado a Cordeau, Gendreau e Laporte (1997)

		Cordeau	AH	
Prob. Teste	Veículos	FO	FO	% Melhora
p02	5	473,87	473,87	0
p03	11	641,20	641,20	0
p04	15	1006,66	1001,59	0,5036
p15	16	4 2534,13	2505,42	1,1329
p19	24	6 3827,06	3860,52	-0,8743
p23	32	9 6139,73	6107.82	0,5197
FO = Distância Total				

parâmetros: na BT nbMax = 1000 e tamList = 150, no AG taxa de cruzamento = 0.9, taxa de mutação = 0.01 e taxa para execução do ABL = 0.02. Para o AH o parâmetro utilizado foi maxHíbrido = 100. O algoritmo foi executado para cada problema teste separadamente, assim como os dados totalizados e armazenados para cada execução. Na Tabela 9, a antepenúltima coluna representa o melhor valor encontrado e a penúltima coluna traz os valores médios para as 50 execuções. A última coluna traz o desvio padrão das 50 execuções. Para um mesmo problema teste foram realizadas 50 execuções, sendo os valores apresentados corresponde à média da FO, que neste experimento é a distância total percorrida, o desvio padrão. O valor da FO não considera aqui o valor do Custo Fixo do veículo, da mesma forma que foi publicado pelo referido autor. Desta forma, foi possível fazer uma comparação entre os resultados do algoritmo AH, proposto neste trabalho, com o de Cordeau, Gendreau e Laporte (1997). Os resultados apresentados aqui mostram que o melhor valor da FO, para cada problema teste, não teve uma variação muito alta em relação ao resultado médio obtido, com as 50 execuções. Os melhores resultados obtidos com o AH, para cada um dos arquivos testes, também estão descritos na Tabela 8.

Tabela 9: Médias obtidas na execução do AH sem Custo Fixo do Veículo

		AH		
Prob. Teste	Clientes	Melhor FO	Média	Desvio
p02	50	473,87	490,75	5,68
p03	75	641,20	719,16	37,19
p04	100	1001,59	1110,29	42,69
p15	160	2505,42	2830,76	191,60
p19	240	3860,52	4156,13	162,30
p23	360	6107.82	6413,59	178.07
FO = Distância Total				

6.2.3.3 Com Múltiplos Depósitos e três Tipos de Veículo

Nesta seção, os experimentos foram realizados com mais de um depósito e com três tipos de veículo. Foi realizada a execução dos algoritmos AG, BT e AH com base nos problemas teste apresentados por Cordeau, Gendreau e Laporte (1997). Foram realizados três tipos de experimentos: o primeiro faz referência apenas à execução do AG. O segundo foi realizado com a execução apenas do algoritmo BT, não considerando o primeiro. Por fim, o terceiro experimento foi realizado com o AH que considera a execução dos dois algoritmos em sequência. Foram considerados múltiplos depósitos e três tipos de veículos. Foram utilizados, de forma empírica, três diferentes valores para o custo fixo de cada um dos tipos de veículo. Nestes experimentos, a FO considera, além da distância total, o custo fixo dos veículos. O primeiro valor utilizado foi o original. O segundo é o valor original, acrescido de 50% deste. Por fim, o terceiro corresponde ao valor original acrescido a 100% dele. Foram utilizados os seguintes parâmetros: na BT nbMax=1000 e tamList=150, no AG taxa de cruzamento = 0.9, taxa de mutação = 0.01 e taxa para execução do ABL = 0.02. Para o AH o parâmetro utilizado foi maxHibrido=100.

Conforme mostrado na Tabela 10, os melhores resultados apresentados na comparação entre algoritmos ocorreram nas execuções do BT e do AH. Pode-se observar, para a mesma solução, que o algoritmo BT contribui mais do que o algoritmo AG. Porém, o algoritmo BT com o AG, compondo o AH, trouxeram um melhor resultado do que executados individualmente. Sobre o custo fixo, o algoritmo sorteia de forma randômica os veículos que compõem a solução. No AG foram 4 do tipo um, 1 do tipo dois e 5 do tipo três com um custo fixo = 720. Nos algoritmos BT e AH foram 1 do tipo um, 2 do tipo dois e 6 do tipo três com um custo fixo = 730.

Tabela 10: Comparação entre AG, BT e AH para o problema teste p04

Resultados para conjunto p04					
	Veículos	Depósitos	Distância	Custo Veículos	Total (FO)
AG	10	2	955,20	720,00	1675,20
BT	9	2	937,25	730,00	1667,25
AH	9	2	934,30	730,00	1664,30
Total (FO) = Distância Total + Custo Fixo dos Veículos					

Na Tabela 11 foi realizada uma simulação com um valor da capacidade do veículo alterado, em relação ao valor original, do problema teste apresentado por Cordeau, Gendreau e Laporte (1997). Foi realizada uma execução com 3 valores, sendo o primeiro o original, conforme mostrado na Tabela 12. Neste experimento, cada execução considerou apenas um valor para a capacidade do veículo. Conforme aumenta a capacidade do veículo, o número de rotas diminui por que veículos maiores diminui a quantidade de viagens. Porém, no caso em que aumentou a capacidade em 100%, não houve a diminuição de 50% do número de veículos. Isso ocorreu porque nem sempre a distância de um cliente até o depósito é a mesma de um cliente para um outro cliente.

Tabela 11: Comportamento da Capacidade dos Tipos de Veículos com o AH

Capacidade do Veículo					
Prob. Teste	Clientes	Capacidade	distância	Qtd Veículos	Depósitos
P02	50	160	473,87	5	4
P02	50	240	479,96	4	4
P02	50	320	471,24	3	4
P03	75	140	641,20	11	5
P03	75	210	639,48	10	5
P03	75	280	624,61	7	5
P04	100	160	1006,66	15	2
P04	100	240	984,51	14	2
P04	100	320	949,61	10	2
P15	160	160	2505,42	16	4
P15	160	240	2448,84	14	4
P15	160	320	2384,64	10	4
P19	240	60	3860,52	24	6
P19	240	90	3792,91	21	6
P19	240	120	3724,63	16	6
P04	360	60	6107,82	32	2
P04	360	90	6010,60	29	2
P04	360	120	5986,46	27	2

Em seguida são apresentados resultados referentes à capacidade e custo dos veículos. Para cada conjunto descrito, o primeiro valor é o original. Para cada execução, um valor do custo fixo é utilizado. Na Tabela 12 são mostrados os diferentes tamanhos da capacidade do veículo e seus custos fixos correspondentes. Foi utilizado o trabalho de Cordeau, Gendreau e Laporte (1997) em que os problemas teste utilizados trouxeram três diferentes valores para a capacidade, 60, 140 e 160. Os valores utilizados nos parâmetros são: nbMax = 750, tamList = 100, numGeração = 300, tamPopulacao = 300, taxaCruzamento = 0,9, taxaMutaçao = 0.02 e maxHibrido = 50.

Para cada conjunto, a primeira linha corresponde ao valor descrito nos arquivos testes. As duas linhas seguintes são valores estabelecidos empiricamente por 50% e 100% do valor original, da capacidade do veículo. Observa-se que, no primeiro exemplo, que é o conjunto p02, o valor original da capacidade do veículo é 160, o valor seguinte é 240 e o terceiro valor é 320. Já a capacidade 140 tem como segundo valor o 210 e o terceiro valor 280. Quanto ao custo Fixo dos veículos, foi considerado que o veículo maior tem um custo maior do que os dois veículos com a metade da capacidade.

A Tabela 13 mostra o resultado de uma simulação com o AH considerando três valores possíveis para a capacidade do veículo, na mesma execução. Os valores correspondem aos três tipos diferentes de veículos abordados neste trabalho. Foi realizada uma execução com 3 valores, conforme é mostrado na Tabela 12. Os valores utilizados nos parâmetros são: nbMax = 750, tamList = 100, numGeração = 300, tamPopulacao = 300, taxaCruzamento = 0,9, taxaMutaçao

Tabela 12: Valores para o Custo Fixo dos Veículos com AH

Veículo com capacidade = 60		
	Capacidade	Custo Fixo
1	60	15
2	90	20
3	120	25
Veículo com capacidade = 140		
	Capacidade	Custo Fixo
1	140	45
2	210	60
3	280	80
Veículo com capacidade = 160		
	Capacidade	Custo Fixo
1	160	50
2	240	70
3	320	90

= 0.02 e maxHíbrido = 50. O algoritmo AH sorteia de forma aleatória o tipo de veículo para cada solução simulada durante a execução. Dos três valores utilizados o primeiro é o original, o segundo valor com 50% do valor do original e por fim, o terceiro valor 100% do valor original.

Tabela 13: Resultado obtidos para Cordeau, Gendreau e Laporte (1997) com Custo Fixo na Função Objetivo

Resultados para conjunto p04 com o AH						
Soluções	Tipo 1	Tipo 2	Tipo 3	Distância	Custo Veículos	Total (FO)
1	2	3	5	949,61	760,00	1709,61
2	4	3	4	959,63	770,00	1729,63
3	1	2	6	937,25	730,00	1667,25
4	5	4	3	949,61	800,00	1749,61
5	10	3	1	1099,68	800,00	1899,68
6	10	3	1	1030,35	800,00	1830,35
7	8	3	2	962,56	790,00	1752,56
8	2	4	6	925,60	920,00	1845,60
9	4	5	3	955,62	820,00	1775,62
10	4	1	5	951,25	720,00	1671,25
Total (FO) = Distância Total + Custo Fixo dos Veículos						

O objetivo foi observar, a partir da referência das instâncias de testes, se o algoritmo conseguiria reduzir em 50% a quantidade dos veículos que a solução original trouxe caso fosse aumentado em 100% o valor da capacidade do veículo. Isso não ocorreu, de forma exata, nem sempre consigo colocar o dobro de clientes em um veículo que antes eram dois em função da restrição da capacidade do veículo. O outro motivo é que mesmo diminuindo idas ao depósito por ter menos veículos, a distância total pode ser maior, porque um cliente que tinha um elo com o depósito, passa a ter um elo com outro cliente e a distância aumentar.

7 CONSIDERAÇÕES FINAIS

7.1 Conclusão

Muitos estudos vêm sendo realizados na construção de algoritmos para a resolução de problemas na área de otimização combinatória. Existem alguns problemas que necessitam de métodos capazes para serem tratados de maneira viável computacionalmente. Estes problemas são de difícil resolução computacional, já que, para avaliação da solução estão envolvidas muitas soluções possíveis. Os métodos exatos não são eficientes, já as heurísticas são métodos mais apropriados para a resolução dos problemas de otimização combinatória, pois se tornam mais eficientes, por não percorrerem todo o espaço de soluções, reduzindo assim o tempo de execução computacional.

Conforme Silva (2010), as empresas necessitam fazer o planejamento do roteiro de entregas dos produtos que estão armazenados em depósitos, com a utilização dos veículos, para atender os pedidos de seus clientes. Necessitam saber de forma eficiente e rápida, qual a maneira mais econômica de montar os roteiros de entregas, que são denominados rotas. A contribuição deste trabalho foi abordar o PRV levando em consideração mais de um depósito e frotas de veículos heterogêneas, de forma simultânea. Além disso foram consideradas restrições como capacidade do veículo, retorno de veículo ao depósito, visita ao cliente apenas uma vez, um cliente ser atendido por apenas um depósito, o limite do número de depósitos, e o limite dos tipos de veículos.

Na BT, foram estudadas diferentes possibilidades de varredura do espaço de soluções, através de diferentes possibilidades de geração de vizinhanças. No AG, foram estudados os operadores para cruzamento parcialmente mapeado (PMX), operador TCX e o operador OX e o operador de mutação troca e o ABL. Por fim, uma implementação híbrida, através do AG e BT. Foi descrita, uma representação matemática referente ao Modelo do Problema de Roteamento de Veículo, considerando Múltiplos Depósitos e Frota Heterogênea.

Na implementação houve a utilização de 5 módulos: Montar a Rede, Gerar Solução Inicial, Algoritmo Híbrido (AH), Algoritmo Genético (AG) e Algoritmo Busca Tabu (BT). Do ponto de vista de implementação o modelo considerando frota heterogênea e depósitos múltiplos, trouxe uma complexidade maior que a maioria dos trabalhos apresentados para o PRV.

Os experimentos puderam ser analisados levando em conta diferentes tamanhos de instâncias de problemas teste, diferentes valores para o nbMax na BT, diferentes valores para a LT na BT, diferentes estratégias de vizinhança utilizando os operadores: troca *swap*, Retira e Insere para intra-rota e inter rota, considerando o mesmo depósito ou com troca de depósito. Já no AG, foram analisadas as variações da Taxa de Cruzamento com 0.80, 0.85 e 0.90, as variações das Taxas de Mutação, entre elas: 0.05, 0.03, 0.02, e 0.01 e a variação do Número de Gerações, entre eles: 100, 250, 500, 750, 1000 e 1500.

Como foi tratado na seção 6.2.1, que tratou do AG, alcançou-se o mesmo número de rotas

e a mesma quantidade de depósitos descrito pelo autor. Além disso, o valor da FO esteve bem próxima do alcançado pelo autor. Conforme os resultados, quanto maior o número de gerações, melhor o resultado da FO. Por fim, havendo a intensificação, ocorre uma melhora do valor da função objetivo. Quanto à estratégia de intensificação utilizada na Busca Local, com a observação de um parâmetro, nota-se que em uma determinada faixa de valor não há a melhora, porém quanto maior o valor de iterações na Busca Local melhor o valor da FO.

Conforme mostrado na seção 6.2.2, pode-se observar que a utilização do movimento Troca obteve resultados melhores do que, o movimento Retira e Insere para geração da vizinhança. Além disso, conforme há um aumento do valor do parâmetro da LT, melhor é o valor da FO. Quanto ao parâmetro nbMax, pode-se observar que quanto maior o seu valor, menor o valor da FO. De acordo com a seção 6.2.3, que faz referência à execução do AH, utilizando o trabalho de Desrochers, Desrosiers e Solomon (1992), que tem apenas um depósito e a capacidade original para os veículos, o objetivo foi comparar o resultado da distância total percorrida, de certa forma não considerando o depósito e a capacidade do veículo. O AH comportou-se de forma satisfatória, alcançando resultados semelhantes ao do autor. Já para o trabalho de Cordeau et al. (2006), os resultados obtidos aqui mostram que o melhor valor da FO para cada problema teste não teve uma variação muito alta em relação ao resultado médio obtido, com as 50 execuções.

Foi realizada uma simulação com a utilização de diferentes valores da capacidade do veículo, em execuções diferentes. Foi possível observar que a maior capacidade do veículo diminuiu o número de rotas. Porém, no caso em que aumentou a capacidade em 100%, não houve a diminuição de 50% do número de veículos. Isso ocorreu, porque nem sempre a distância de um cliente até o depósito é a mesma de um cliente para um outro cliente e porque o rearranjo dos clientes em um veículo, que antes eram atendidos por dois veículos, pode ferir a restrição da capacidade do veículo.

7.2 Trabalhos Futuros

Nesta seção serão apresentados alguns tópicos de possíveis assuntos que poderiam dar continuidade a este trabalho, bem como proporcionar novas pesquisas para a contribuição da ciência em especial na área de pesquisa operacional e otimização combinatória. Realizar estudos que possam aprimorar os conhecimentos nos diferentes tipos e variações do PRV. Desenvolver um operador de cruzamento com uma técnica nova que seja adequada ao PRVC. O operador deverá contemplar alguma das variações do PRV. Estudar novas meta-heurísticas apropriadas ao PRV. Propor uma meta-heurística capaz de solucionar o problema do PRV. Estudar especificamente o problema do carregamento de veículos. Propor um heurísticas adequada a resolver o problema de carregamento de veículos.

REFERÊNCIAS

- AFSHAR-NADJAFI, B.; AFSHAR-NADJAFI, A. A constructive heuristic for time-dependent multi-depot vehicle routing problem with time-windows and heterogeneous fleet. **Journal of King Saud University-Engineering Sciences**, [S.l.], 2014.
- ANBUUDAYASANKAR, S.; GANESH, K.; KOH, S. L.; DUCQ, Y. Modified savings heuristics and genetic algorithm for bi-objective vehicle routing problem with forced backhauls. **Expert Systems with Applications**, [S.l.], v. 39, n. 3, p. 2296–2305, 2012.
- AUGERAT, P.; BELENGUER, J. M.; BENAVENT, E.; CORBÉLAN, A.; NADDEF, D. Separating capacity constraints in the CVRP using tabu search. **European Journal of Operational Research**, [S.l.], v. 106, n. 2, p. 546–557, 1998.
- BAKER, B. M.; AYECHHEW, M. A genetic algorithm for the vehicle routing problem. **Computers & Operations Research**, [S.l.], v. 30, n. 5, p. 787–800, 2003.
- BELHAIZA, S.; HANSEN, P.; LAPORTE, G. A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. **Computers & Operations Research**, [S.l.], v. 52, p. 269–281, 2014.
- BERGER, J.; BARKAOUI, M. A hybrid genetic algorithm for the capacitated vehicle routing problem. In: GENETIC AND EVOLUTIONARY COMPUTATION—GECCO 2003, 2003. **Anais...** [S.l.: s.n.], 2003. p. 646–656.
- BRANDÃO, J. A tabu search algorithm for the open vehicle routing problem. **European Journal of Operational Research**, [S.l.], v. 157, n. 3, p. 552–564, 2004.
- BRANDÃO, J. A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem. **European journal of operational research**, [S.l.], v. 195, n. 3, p. 716–728, 2009.
- BRANDÃO, J. A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. **Computers & Operations Research**, [S.l.], v. 38, n. 1, p. 140–151, 2011.
- BRANDAO, J.; MERCER, A. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. **European journal of operational research**, [S.l.], v. 100, n. 1, p. 180–191, 1997.
- BRÄYSY, O.; GENDREAU, M. Tabu search heuristics for the vehicle routing problem with time windows. **Top**, [S.l.], v. 10, n. 2, p. 211–237, 2002.
- CESCHIA, S.; DI GASPERO, L.; SCHAERF, A. Tabu search techniques for the heterogeneous vehicle routing problem with time windows and carrier-dependent costs. **Journal of Scheduling**, [S.l.], v. 14, n. 6, p. 601–615, 2011.
- CHENG, C.-B.; WANG, K.-P. Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm. **Expert Systems with Applications**, [S.l.], v. 36, n. 4, p. 7758–7763, 2009.
- CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. **Operations research**, [S.l.], v. 12, n. 4, p. 568–581, 1964.

CORDEAU, J.-F.; GENDREAU, M.; LAPORTE, G. A tabu search heuristic for periodic and multi-depot vehicle routing problems. **Networks**, [S.l.], v. 30, n. 2, p. 105–119, 1997.

CORDEAU, J.-F.; GENDREAU, M.; LAPORTE, G.; POTVIN, J.-Y.; SEMET, F. A guide to vehicle routing heuristics. **Journal of the Operational Research society**, [S.l.], p. 512–522, 2002.

CORDEAU, J.-F.; LAPORTE, G.; SAVELSBERGH, M. W.; VIGO, D. Vehicle routing. **Transportation, handbooks in operations research and management science**, [S.l.], v. 14, p. 367–428, 2006.

CRINO, J.; MOORE, J.; BARNES, J.; NANRY, W. Solving the theater distribution vehicle routing and scheduling problem using group theoretic tabu search. **Mathematical and computer modelling**, [S.l.], v. 39, n. 6, p. 599–616, 2004.

DAYARIAN, I.; CRAINIC, T. G.; GENDREAU, M.; REI, W. A branch-and-price approach for a multi-period vehicle routing problem. **Computers & Operations Research**, [S.l.], v. 55, p. 167–184, 2015.

DESROCHERS, M.; DESROSIERS, J.; SOLOMON, M. A new optimization algorithm for the vehicle routing problem with time windows. **Operations research**, [S.l.], v. 40, n. 2, p. 342–354, 1992.

ESCOBAR, J. W.; LINFATI, R.; TOTH, P.; BALDOQUIN, M. G. A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem. **Journal of Heuristics**, [S.l.], v. 20, n. 5, p. 483–509, 2014.

EUCHI, J.; CHABCHOUB, H. A hybrid tabu search to solve the heterogeneous fixed fleet vehicle routing problem. **Logistics Research**, [S.l.], v. 2, n. 1, p. 3–11, 2010.

FLESZAR, K.; OSMAN, I. H.; HINDI, K. S. A variable neighbourhood search algorithm for the open vehicle routing problem. **European Journal of Operational Research**, [S.l.], v. 195, n. 3, p. 803–809, 2009.

GALAFASSI, C. **Aplicação de Metaheurística na Abordagem do Problema de Roteamento de Veículos Capacitados com Janelas de Tempo**. 2011. Dissertação de Mestrado em Computação Aplicada — Universidade do Vale do Rio dos Sinos, Porto Alegre, 2011.

GHOSEIRI, K.; GHANNADPOUR, S. F. Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. **Applied Soft Computing**, [S.l.], v. 10, n. 4, p. 1096–1107, 2010.

GLOVER, F.; LAGUNA, M. **Tabu Search**. [S.l.]: Springer, 2013.

GLOVER, F.; TAILLARD, E. A user's guide to tabu search. **Annals of operations research**, [S.l.], v. 41, n. 1, p. 1–28, 1993.

GLOVER, F. W.; KOCHENBERGER, G. A. **Handbook of metaheuristics**. [S.l.]: Springer Science & Business Media, 2006. v. 57.

HENKE, T.; SPERANZA, M. G.; WÄSCHER, G. The multi-compartment vehicle routing problem with flexible compartment sizes. **European Journal of Operational Research**, [S.l.], v. 246, n. 3, p. 730–743, 2015.

- HERRERO, R.; RODRÍGUEZ, A.; CÁCERES-CRUZ, J.; JUAN, A. A. Solving vehicle routing problems with asymmetric costs and heterogeneous fleets. **International Journal of Advanced Operations Management**, [S.l.], v. 6, n. 1, p. 58–80, 2014.
- HO, S. C.; HAUGLAND, D. A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. **Computers & Operations Research**, [S.l.], v. 31, n. 12, p. 1947–1964, 2004.
- HO, W.; HO, G. T.; JI, P.; LAU, H. C. A hybrid genetic algorithm for the multi-depot vehicle routing problem. **Engineering Applications of Artificial Intelligence**, [S.l.], v. 21, n. 4, p. 548–557, 2008.
- HOLLAND, J. H. **Adaptation in natural and artificial systems**: an introductory analysis with applications to biology, control, and artificial intelligence. [S.l.]: U Michigan Press, 1975.
- HWANG, H.-S. An improved model for vehicle routing problem with time constraint based on genetic algorithm. **Computers & Industrial Engineering**, [S.l.], v. 42, n. 2, p. 361–369, 2002.
- JEON, G.; LEEP, H. R.; SHIM, J. Y. A vehicle routing problem solved by using a hybrid genetic algorithm. **Computers & Industrial Engineering**, [S.l.], v. 53, n. 4, p. 680–692, 2007.
- KUMAR, S. N.; PANNEERSELVAM, R. A survey on the vehicle routing problem and its variants. **Intelligent Information Management**, [S.l.], v. 4, n. 3, p. 66, 2012.
- LI, F.; GOLDEN, B.; WASIL, E. The open vehicle routing problem: algorithms, large-scale test problems, and computational results. **Computers & operations research**, [S.l.], v. 34, n. 10, p. 2918–2930, 2007.
- LI, X.; LEUNG, S. C.; TIAN, P. A multistart adaptive memory-based tabu search algorithm for the heterogeneous fixed fleet open vehicle routing problem. **Expert Systems with Applications**, [S.l.], v. 39, n. 1, p. 365–374, 2012.
- LI, X.; TIAN, P.; LEUNG, S. C. Vehicle routing problems with time windows and stochastic travel and service times: models and algorithm. **International Journal of Production Economics**, [S.l.], v. 125, n. 1, p. 137–145, 2010.
- MARINAKIS, Y.; MIGDALAS, A.; PARDALOS, P. M. A new bilevel formulation for the vehicle routing problem and a solution method using a genetic algorithm. **Journal of Global Optimization**, [S.l.], v. 38, n. 4, p. 555–580, 2007.
- MIRABI, M. A novel hybrid genetic algorithm for the multidepot periodic vehicle routing problem. **Artificial Intelligence for Engineering Design, Analysis and Manufacturing**, [S.l.], v. 29, n. 01, p. 45–54, 2015.
- MONTANÉ, F. A. T.; GALVAO, R. D. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. **Computers & Operations Research**, [S.l.], v. 33, n. 3, p. 595–619, 2006.
- NGUYEN, P. K.; CRAINIC, T. G.; TOULOUSE, M. A hybrid generational genetic algorithm for the periodic vehicle routing problem with time windows. **Journal of Heuristics**, [S.l.], v. 20, n. 4, p. 383–416, 2014.

- PAPADIMITRIOU, C. H.; STEIGLITZ, K. **Combinatorial optimization**: algorithms and complexity. [S.l.]: Courier Corporation, 1982.
- PISINGER, D.; ROPKE, S. A general heuristic for vehicle routing problems. **Computers & operations research**, [S.l.], v. 34, n. 8, p. 2403–2435, 2007.
- POTVIN, J.-Y.; DUHAMEL, C.; GUERTIN, F. A genetic algorithm for vehicle routing with backhauling. **Applied Intelligence**, [S.l.], v. 6, n. 4, p. 345–355, 1996.
- REITER, P.; GUTJAHR, W. J. Exact hybrid algorithms for solving a bi-objective vehicle routing problem. **Central European Journal of Operations Research**, [S.l.], v. 20, n. 1, p. 19–43, 2012.
- ROCHAT, Y.; TAILLARD, É. D. Probabilistic diversification and intensification in local search for vehicle routing. **Journal of heuristics**, [S.l.], v. 1, n. 1, p. 147–167, 1995.
- SILVA, M. R. **Uma contribuição ao projeto de redes de transporte de carga parcelada**. 2010. Tese (Doutorado em Ciência da Computação) — Universidade de São Paulo, 2010.
- SIMAS, E. P. L.; GÓMEZ, A. T. Comparing a tabu search process-using and not using and intensification strategy to solve the vehicle routing problem. In: ICINCO-ICSO, 2007. **Anais...** [S.l.: s.n.], 2007. p. 77–84.
- SUBRAMANIAN, A.; PENNA, P. H. V.; UCHOA, E.; OCHI, L. S. A hybrid algorithm for the heterogeneous fleet vehicle routing problem. **European Journal of Operational Research**, [S.l.], v. 221, n. 2, p. 285–295, 2012.
- TAN, K. C.; CHEONG, C. Y.; GOH, C. K. Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. **European Journal of operational research**, [S.l.], v. 177, n. 2, p. 813–839, 2007.
- TOTH, P.; VIGO, D. A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls. **European Journal of Operational Research**, [S.l.], v. 113, n. 3, p. 528–543, 1999.
- URSANI, Z.; ESSAM, D.; CORNFORTH, D.; STOCKER, R. Localized genetic algorithm for vehicle routing problem with time windows. **Applied Soft Computing**, [S.l.], v. 11, n. 8, p. 5375–5390, 2011.
- WANG, Z.; LI, Y.; HU, X. A heuristic approach and a tabu search for the heterogeneous multi-type fleet vehicle routing problem with time windows and an incompatible loading constraint. **Computers & Industrial Engineering**, [S.l.], v. 89, p. 162–176, 2015.
- WASSAN, N. A.; WASSAN, A. H.; NAGY, G. A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries. **Journal of combinatorial optimization**, [S.l.], v. 15, n. 4, p. 368–386, 2008.
- YUAN, S.; SKINNER, B.; HUANG, S.; LIU, D. A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. **European Journal of Operational Research**, [S.l.], v. 228, n. 1, p. 72–82, 2013.
- YÜCENUR, G. N.; DEMIREL, N. Ç. A new geometric shape-based genetic clustering algorithm for the multi-depot vehicle routing problem. **Expert Systems with Applications**, [S.l.], v. 38, n. 9, p. 11859–11865, 2011.