CAIO DOMINGUES REINA

ROTEIRIZAÇÃO DE VEÍCULOS COM JANELAS DE TEMPO UTILIZANDO ALGORITMO GENÉTICO

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO DEPARTAMENTO DE ENGENHARIA DE TRANSPORTES

CAIO DOMINGUES REINA

ROTEIRIZAÇÃO DE VEÍCULOS COM JANELAS DE TEMPO UTILIZANDO ALGORITMO GENÉTICO

CAIO DOMINGUES REINA

ROTEIRIZAÇÃO DE VEÍCULOS COM JANELAS DE TEMPO UTILIZANDO ALGORITMO GENÉTICO

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia.

CAIO DOMINGUES REINA

ROTEIRIZAÇÃO DE VEÍCULOS COM JANELAS DE TEMPO UTILIZANDO ALGORITMO GENÉTICO

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia.

Área de Concentração: Engenharia de Transportes

Orientador: Prof. Dr. Edvaldo Simões da Fonseca Junior

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.
São Paulo, de junho de 2012.
Assinatura do autor
Assinatura do orientador

FICHA CATALOGRÁFICA

Reina, Caio Domingues

Roteirização de veículos com janelas de tempo utilizando algoritmo genético / C.D. Reina. -- ed.rev. -- São Paulo, 2012.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Transportes.

1.Transporte rodoviário 2.Roteirização 3.Algoritmos geneticos I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Transportes II.t.

Dedico,

À minha mãe, ao meu pai e ao meu irmão pelo incentivo e por serem meu porto seguro.

À minha esposa por todo amor, carinho e apoio.

AGRADECIMENTOS

Ao Prof. Dr. Edvaldo Simões da Fonseca Jr. pela oportunidade, orientação e direcionamento dessa pesquisa.

À Profa. Silvely Nogueira de Almeida Salomão Néia pelo auxílio no desenvolvimento dessa pesquisa, e principalmente por ter me apresentado à otimização, me apoiando desde o início.

Ao Prof. Dr. Cláudio Barbieri da Cunha pelos conselhos transmitidos em relação à abordagem do problema e pelas sugestões tão pertinentes na criação do algoritmo desenvolvido.

Aos professores da Escola Politécnica pela oportunidade de aprendizado.

Aos meus queridos pais Valmir Atiensa Reina e Antonina Domingues de Oliveira Reina e irmão Thiago Domingues Reina pela força e pelo amor incondicional.

À minha esposa Michelly Lima Reina por sempre acreditar, por todo sentimento de otimismo transmitido e pela compreensão em relação ao tempo em que teve que dividir minha atenção com essa pesquisa.

Aos velhos amigos Pedro Victor Losada Cavalcante, Raphael Henrique Rinaldi e Vitor Shigueyoshi Takahara Oda pela amizade tão valiosa há tantos anos.

Aos amigos Gabriel do Nascimento Guimarães, Wagner Carrupt Machado, Luiz Felipe Sartori Gonçalves, Dino Sérgio Tomazo Lorenzi, Anderson Morais Mori e aos demais integrantes do grupo GIGA pelo apoio em cada passo dessa caminhada.

A todos os amigos que de alguma forma colaboraram com essa pesquisa.

RESUMO

O componente de planejamento faz parte do projeto de desenvolvimento dos veículos autônomos, e é responsável por gerar rotas para o sistema como um todo. Em aplicações em que o veículo deve visitar pontos em intervalos de tempo pré-determinados, o componente de planejamento se enquadra em um problema de roteirização conhecido da literatura, denominado problema de roteirização de veículos com janelas de tempo. Tal problema é uma generalização do problema clássico de roteirização de veículos classificado no grupo de problemas NP-Hard. Esse trabalho apresenta uma proposta de solução para o problema baseada na metaheurística algoritmo genético. Os cromossomos foram representados pela ordem de atendimento dos clientes sem delimitadores de rota. Para quebrar os cromossomos em rotas, foi utilizado um procedimento adaptado baseado em Prins (2004). A população inicial se constitui por uma parte construída com cromossomos criados aleatoriamente e outra parte construída através da heurística de inserção II de Solomon (1987), com quatro formas diferentes de inserir o primeiro cliente de cada rota. Na fase de recombinação, foram utilizados quatro tipos de crossover: uniforme, dois pontos, heurístico e PMX, e um operador de mutação baseado em uma busca heurística. A cada geração foram aplicados princípios de elitismo e pós-otimização utilizando a heurística λ-interchange de Osman (1993). O algoritmo foi testado nos conjuntos C1, C2, R1, R2, RC1 e RC2 de Solomon (1987) e os resultados foram comparados com os melhores resultados encontrados na literatura.

Palavras-chave: transporte rodoviário, roteirização, algoritmos genéticos.

ABSTRACT

The planning component is a part of autonomous vehicle development project and it is responsible to whole. In applications which vehicle must to visit way points generate routes for the system as a at predetermined intervals of time, the planning component fits into a routing problem known in the literature called routing problem with time windows. This problem is a generalization of the classical vehicle routing problem classified in the group of NP- Hard problems. This thesis presents a solution proposal to problem based on genetic algorithm metaheuristic. Chromosomes were represented by the order of serving customers without delimiters route. To split the chromosomes on routes, it is used a procedure adapted based on Prins (2004). The initial population is constituted by two parts: one with randomly created chromosomes and another constructed through the insertion heuristic I1 of Solomon (1987), with four different ways of insertion of the first customer of each route. In the recombination step, four types of crossover were used: uniform, two points, heuristic, and PMX, and a mutation operator based on heuristic search. In each generation it is applied principles of elitism and postoptimization using the λ -interchange heuristic of Osman (1993). The algorithm was tested on the sets C1, C2, R1, R2, RC1 and RC2 of Solomon (1987) and the results were compared with the best results found in the literature.

Keywords: road transport, routing, genetic algorithms.

SUMÁRIO

1.	INTRODUÇÃO	1
	1.1. Objetivo	2
	1.2. VEÍCULO AUTÔNOMO	3
	1.2.1. Pesquisa sobre veículos autônomos dentro do grupo GIGA	5
	1.3. ESTRUTURA DO ALGORITMO	6
	1.4. DELINEAMENTO DO TRABALHO	7
2.	REVISÃO BIBLIOGRÁFICA	8
	2.1. TAXONOMIA DOS PROBLEMAS DE ROTEIRIZAÇÃO DE VEÍCULOS	8
	2.1. MÉTODOS EXATOS	11
	2.2. Heurísticas	13
	2.3. Metaheurísticas	15
	2.4. MELHORES RESULTADOS DA LITERATURA PARA O VRPTW	20
3.	CARACTERIZAÇÃO DO PROBLEMA	24
	3.1. O PROBLEMA CLÁSSICO DE ROTEIRIZAÇÃO DE VEÍCULOS	24
	3.2. PROBLEMA DE ROTEIRIZAÇÃO DE VEÍCULOS COM JANELA DE TEMPO	26
4.	ESTRATÉGIA DE SOLUÇÃO	28
	4.1. ALGORITMO GENÉTICO	28
	4.2. ALGORITMO PROPOSTO	30
	4.2.1 Geração da população inicial	31
	4.2.1.1 Push Forward Insertion Heuristic	31
	4.2.1.2 Algoritmo Split	35
	4.3. Seleção	40
	4.4. Avaliação	41
	4.5. Operadores Genéticos	42
	4.5.1. <i>Crossover</i>	43
	4.6. ELITISMO	49
	4.7. Pós-Otimização	50
	4.7.1. λ-interchange	50
	4.8. DESCRIÇÃO DOS PARÂMETROS UTILIZADOS	51
5.	EXPERIMENTOS COMPUTACIONAIS	55
	5.1. INSTÂNCIAS DE SOLOMON (1987)	55

5.2. Perfis de Desempenho	55
5.3. DEFINIÇÃO DOS VALORES DOS PARÂMETROS UTILIZADOS NO ALGORITMO	56
5.3.1. <i>Crossover</i>	57
5.3.2. Mutação	63
5.3.3. Push Forward Insertion Heuristic	66
5.3.4. Função de Avaliação	FINIÇÃO DOS VALORES DOS PARÂMETROS UTILIZADOS NO ALGORITMO
5.4. AVALIAÇÃO DO ALGORITMO PROPOSTO	74
6. CONSIDERAÇÕES FINAIS	79
6.1. Conclusões	79
6.2. RECOMENDAÇÕES E PRÓXIMAS ETAPAS	82
7. REFERÊNCIAS BIBLIOGRÁFICAS	84

LISTA DE FIGURAS

FIGURA 1.1 - VEÍCULOS VENCEDORES DAS TRÊS EDIÇÕES DO DARPA. STANDSTORM (EDIÇÃO)
2004) À ESQUERDA, STANLEY (EDIÇÃO 2005) NO CENTRO E BOSS (EDIÇÃO 2007) À DIREITA	3
FIGURA 1.2 - DIAGRAMA DE RELACIONAMENTO ENTRE OS DIFERENTES COMPONENTES FUNCIO	ONAIS
DE UM VEÍCULO AUTÔNOMO (TRADUZIDO DE WHYTE, 2001)	4
FIGURA 1.3 - VEÍCULO AUTÔNOMO DESENVOLVIDO POR GONÇALVES (2011)	5
FIGURA 1.4 - PLATAFORMA ADQUIRIDA PELO GRUPO GIGA PARA A PESQUISA DE VEÍCULOS	
AUTÔNOMOS	6
FIGURA 4.1 - FLUXOGRAMA DOS PRINCIPAIS PASSOS DO ALGORITMO GENÉTICO (LINDEN, 200	6)30
FIGURA 4.2 - ALGORITMO PFIH DESENVOLVIDO	35
FIGURA 4.3 - EXEMPLO DA CRIAÇÃO DE UM CROMOSSOMO. FONTE: PRINS (2004), P. 1989	37
FIGURA 4.4 - PSEUDOCÓDIGO DO ALGORITMO DE ROTULAÇÃO, PRINS (2004)	38
FIGURA 4.5 - PSEUDOCÓDIGO DO ALGORITMO DE CONSTRUÇÃO, PRINS (2004)	38
FIGURA 4.6 - PSEUDOCÓDIGO DO ALGORITMO DE ROTULAÇÃO, CHENG (2005)	39
FIGURA 4.7 - PSEUDOCÓDIGO DO ALGORITMO DE ROTULAÇÃO UTILIZADO NESSA PESQUISA	40
FIGURA 4.8 - MÉTODO DA ROLETA VICIADA PARA ESCOLHA DO CROSSOVER	43

LISTA DE TABELAS

TABELA 2.1 - TAXONOMIA DO PROBLEMA DE ROTEIRIZAÇÃO DE VEÍCULOS SEGUNDO EKSIOGLU E	ΞT
AL. (2009)	9
TABELA 2.2 - LEVANTAMENTO DOS MELHORES RESULTADOS PUBLICADOS NA LITERATURA	
ENVOLVENDO MÉTODOS EXATOS E HEURÍSTICOS, APLICADOS AO VRPTW	.21
TABELA 4.1 - PARÂMETROS CONSIDERADOS NO ALGORITMO DESENVOLVIDO	.51
Tabela 5.1 - Parâmetros utilizados nos testes para definição das taxas dos	
CROSSOVERS	.58
TABELA 5.2 - CONFIGURAÇÕES DAS TAXAS DE CROSSOVER UTILIZADAS NOS TESTES	.59
TABELA 5.3 - CONFIGURAÇÕES DAS TAXAS DE MUTAÇÃO UTILIZADAS NOS TESTES	.63
TABELA 5.4 - CONFIGURAÇÕES DAS TAXAS DE PFIH UTILIZADAS NOS TESTES	.67
TABELA 5.5 - CONFIGURAÇÕES DOS PESOS DA FUNÇÃO DE AVALIAÇÃO UTILIZADA NOS TESTES	.71
TABELA 5.6 - VALORES DOS PARÂMETROS UTILIZADOS NO ALGORITMO PROPOSTO	.74
TABELA 5.7 - RESULTADOS OBTIDOS UTILIZANDO O ALGORITMO PROPOSTO E COMPARAÇÃO COM	[
OS MELHORES RESULTADOS PUBLICADOS NA LITERATURA	.75
TABELA 5.8 - MÉDIA POR CLASSE DOS RESULTADOS OBTIDOS E DOS MELHORES RESULTADOS	
PUBLICADOS	.77

LISTA DE QUADROS

QUADRO 5.1 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DAS TAXAS DOS	
CROSSOVERS PARA OS PROBLEMAS DO GRUPO C1	60
QUADRO 5.2 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DAS TAXAS DOS	
CROSSOVERS PARA OS PROBLEMAS DO GRUPO C2	60
QUADRO 5.3 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DAS TAXAS DOS	
CROSSOVERS PARA OS PROBLEMAS DO GRUPO R1	61
QUADRO 5.4 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DAS TAXAS DOS	
CROSSOVERS PARA OS PROBLEMAS DO GRUPO R1	61
QUADRO 5.5 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DAS TAXAS DOS	
CROSSOVERS PARA OS PROBLEMAS DO GRUPO RC1	62
QUADRO 5.6 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DAS TAXAS DOS	
CROSSOVERS PARA OS PROBLEMAS DO GRUPO RC2	62
QUADRO 5.7 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DA TAXA DE MUTAÇÃO	
PARA OS PROBLEMAS DO GRUPO C1	63
QUADRO 5.8 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DA TAXA DE MUTAÇÃO	
PARA OS PROBLEMAS DO GRUPO C2	64
QUADRO 5.9 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DA TAXA DE MUTAÇÃO	
PARA OS PROBLEMAS DO GRUPO R1	64
QUADRO 5.10 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DA TAXA DE MUTAÇÃO	
PARA OS PROBLEMAS DO GRUPO R2	65
QUADRO 5.11 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DA TAXA DE MUTAÇÃO	
PARA OS PROBLEMAS DO GRUPO RC1	65
QUADRO 5.12 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DA TAXA DE MUTAÇÃO	
PARA OS PROBLEMAS DO GRUPO RC2	66
QUADRO 5.13 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DAS TAXAS DA	
HEURÍSTICA DE CONSTRUÇÃO PFIH PARA OS PROBLEMAS DO GRUPO C1	67
QUADRO 5.14 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DAS TAXAS DA	
HEURÍSTICA DE CONSTRUÇÃO PFIH PARA OS PROBLEMAS DO GRUPO C2	68
QUADRO 5.15 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DAS TAXAS DA	
HEURÍSTICA DE CONSTRUÇÃO PFIH PARA OS PROBLEMAS DO GRUPO R1	68
Quadro 5.16 - Curvas de perfil de desempenho para definição das taxas da	
HEURÍSTICA DE CONSTRUÇÃO PEIH PARA OS PROBIEMAS DO GRUPO R2	69

QUADRO 5.17 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DAS TAXAS DA	
HEURÍSTICA DE CONSTRUÇÃO PFIH PARA OS PROBLEMAS DO GRUPO RC1	69
QUADRO 5.18 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DAS TAXAS DA	
HEURÍSTICA DE CONSTRUÇÃO PFIH PARA OS PROBLEMAS DO GRUPO RC2	70
Q UADRO 5.19 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DOS PESOS DA FUNÇÃO DE	
AVALIAÇÃO PARA OS PROBLEMAS DO GRUPO C1	71
Q UADRO 5.20 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DOS PESOS DA FUNÇÃO DE	
AVALIAÇÃO PARA OS PROBLEMAS DO GRUPO C2	71
Q UADRO 5.21 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DOS PESOS DA FUNÇÃO DE	
AVALIAÇÃO PARA OS PROBLEMAS DO GRUPO R1.	72
Q UADRO 5.22 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DOS PESOS DA FUNÇÃO DE	
AVALIAÇÃO PARA OS PROBLEMAS DO GRUPO R2	72
Q UADRO 5.23 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DOS PESOS DA FUNÇÃO DE	
AVALIAÇÃO PARA OS PROBLEMAS DO GRUPO RC1	73
Q UADRO 5.24 - CURVAS DE PERFIL DE DESEMPENHO PARA DEFINIÇÃO DOS PESOS DA FUNÇÃO DE	
AVALIAÇÃO PARA OS PROBLEMAS DO GRUPO RC2	73

LISTA DE ABREVIATURAS E SIGLAS

ABA Active Brake Assist

DARPA Defense Advanced Research Projects Agency

DAS Driver Assistance Systems

DoD Department of Defense

FB First Best
GB Global Best

GIGA Grupo de Investigação em Geomática Aplicada à engenharia

GNSS Global Navigation Satellite System

GPS Global Positioning System

GRASP Greedy Randomized Adaptive Search Procedure

LDWS Lane Departure Warning System

LNS Large Neighbourhood Search

OX order crossover

PFIH Push Forward Insertion Heuristic

PFIH-AL Push Forward Insertion Heuristic com cliente inicial aleatório

Push Forward Insertion Heuristic com cliente inicial sendo o mais distante do

PFIH-DD depósito

Push Forward Insertion Heuristic com cliente inicial sendo escolhido através da PFIH-EQ

equação ponderada baseada na equação apresentada por Thangiah *et al.* (1994)

Push Forward Insertion Heuristic com cliente inicial sendo o que apresenta menor

PFIH-JT instante final da janela de tempo

PMX Partially Mapped Crossover

VANT Veículo Aéreo Não Tripulado

VRP Vehicle Routing Problem

VRPTW Vehicle Routing Problem With Time Windows

1. INTRODUÇÃO

O aumento do número de veículos em grandes centros tem trazido problemas para os cidadãos com respeito a congestionamentos, a poluição e a segurança. Segundo Benenson (2008), com o desenvolvimento e a implantação de veículos autônomos é possível melhorar o tráfego nas cidades, reduzir o número de acidentes, reduzir a poluição e, sobretudo melhorar a qualidade de vida da população.

Diversos fabricantes de veículos e a comunidade científica realizam esforços para que a tecnologia empregada nos veículos urbanos possibilite uma condução cada vez mais apoiada por sistemas eletrônicos de alta precisão denominados Sistemas de Assistência ao Condutor (*DAS*, *Driver Assistance Systems*). Algumas dessas inovações já estão embutidas em veículos comercializados, como o Assistente Ativo de Frenagem (*ABA*, *Active Brake Assist*) que é um sistema de identificação de colisão e frenagem implantado em alguns modelos de ônibus e caminhões da Mercedez-Benz e o Sistema de Aviso de Desvio de Rota (*LDWS*, *Lane Departure Warning*) que utiliza visão computacional para identificar se o veículo está saindo dos limites das faixas da pista (Gonçalves, 2011). Espera-se que, no médio e longo prazo, essa tecnologia evolua de tal maneira que viabilize a navegação totalmente autônoma de veículos em ambientes urbanos.

Em paralelo, existe outra linha de desenvolvimento de veículos autônomos com aplicações específicas, em que todo o projeto de concepção do veículo está ligado ao uso final do mesmo. Alguns exemplos dessas aplicações são os robôs de coleta de lixo, os aspiradores de pó autônomos, VANTs (veículo aéreo não tripulado), coleta e entrega de peças dentro da área de produção em indústrias, vigilância, transporte de passageiros dentro de aeroportos, veículos exploradores, dentre uma infinidade de outras possíveis aplicações.

Com o objetivo de explorar o potencial uso de veículos autônomos em diferentes aplicações e dominar a tecnologia envolvida, o grupo GIGA (Grupo de Investigação e, Geomática Aplicada à Engenharia) do Departamento de Engenharia de Transportes – EPUSP (Escola Politécnica da USP) atua em uma linha de pesquisa para estudar e projetar a navegação de veículos autônomos. Essa pesquisa está inserida na linha de pesquisa do GIGA com o objetivo de criar um algoritmo eficaz de planejamento de rotas para um conjunto de veículos, de forma que os mesmos realizem visitas em pontos específicos dentro de janelas de tempo estipuladas.

O desafio de criar rotas onde os veículos devem realizar visitas em pontos específicos é um problema logístico denominado problema de roteirização de veículos (*VRP*, *vehicle routing problem*), e é estudado há muito tempo pela comunidade científica. O *VRP* pode ser definido, em linhas gerais, como um problema de distribuição onde se busca encontrar rotas econômicas que atendam um dado conjunto de clientes. A distribuição é feita a partir de um depósito central, utilizando uma frota de veículos com capacidades limitadas que devem passar pelos clientes, satisfazendo integralmente suas demandas, e retornar ao depósito.

Devido à complexidade computacional do *VRP* em função de sua natureza combinatória, o problema se encaixa na categoria de problemas NP-*Hard* (Bodin *et al.*, 1983). Para tais problemas não se conhece um algoritmo polinomial que os resolva com garantia de solução ótima , ao invés disso, o tempo de resolução é exponencial. Sendo assim, dependendo do tamanho do problema, não existe um algoritmo exato que garanta solução ótima em tempo hábil, pois o esforço computacional para sua resolução cresce de forma exponencial com o número de clientes a serem atendidos.

O *VRP* clássico foi introduzido por Dantzig e Ramser (1959), desde então muitas variações são propostas de modo a incorporar as características dos problemas encontrados em complexas redes de distribuição. Uma dessas variações é a adição da restrição de janelas de tempo para os clientes, configurando assim o problema de roteirização de veículos com janelas de tempo (*VRPTW*, *vehicle routing problem with time windows*) que se enquadrou perfeitamente no contexto de pesquisa desse trabalho. Essa nova restrição obriga os veículos a visitarem os clientes dentro de janelas de horário predefinidas. Segundo Solomon e Desrosiers (1988), o problema de roteirização de veículos com janelas de tempo continua sendo um problema NP- *Hard* por ser extensão do *VRP*.

1.1. Objetivo

O objetivo desse trabalho é apresentar uma proposta de solução para o *VRPTW* para um conjunto de veículos autônomos de forma que os mesmos recebam missões específicas de pontos a serem visitados dentro de uma janela de tempo estipulada, minimizando a distância total percorrida.

O algoritmo será baseado na metaheurística algoritmo genético e as soluções obtidas serão testadas nas instâncias de Solomon (1987) e comparadas com as soluções obtidas nos melhores resultados publicados na literatura.

1.2. Veículo Autônomo

Os veículos autônomos são capazes de operar em diferentes tipos de ambientes sem a necessidade da intervenção humana direta. Para tanto, eles são equipados com diversos instrumentos sensores que capturam informações do ambiente em que estão inseridos para desenvolvimento de um mapeamento que os auxiliam na tomada de decisão.

Atualmente, os veículos autônomos mais evoluídos, destinados às operações em centros urbanos, participam de uma competição organizada pela agência de pesquisa do departamento de defesa dos EUA (DoD, *Department of Defense*) dos EUA denominada DARPA (*Defense Advanced Research Projects Agency*) cujo objetivo é desenvolver tecnologia autônoma de veículos militares. O DARPA *Gran Challenge*, como é chamada a competição, já possui três edições realizadas nos anos 2004, 2005 e 2007.

Na primeira edição do DARPA, o objetivo era cumprir um trajeto de 240 km no deserto de Mojave nos EUA. Nenhum dos 13 veículos conseguiu finalizar a prova, sendo que o veículo SandStorm desenvolvido pela equipe *Red Team*, da Universidade Carnegie Mellon foi considerado o vencedor da prova com 12 km. Na segunda edição, apenas 5 dos 23 veículos conseguiram completar a prova, também realizada em terreno *off-road* com distância total de 212 km, o veículo vencedor foi o Stanley da equipe *Stanford Racing Team* da Universidade de Stanford, com tempo de 06h 54min. A terceira edição foi marcada por mudanças significativas em relação às duas edições anteriores, dessa vez foi adotado um ambiente urbano onde os veículos deveriam visitar pontos estabelecidos respeitando as leis de trânsito local e evitando colisões com outros veículos. Nesta edição 6 dos 11 veículos participantes completaram a prova sendo que o vencedor foi o veículo *Boss* da equipe *Tartan Racing* desenvolvido por uma parceria entre a Universidade Carnegie Mellon e a General Motors Corporation, concluindo a prova em 04h 10min. Os veículos vencedores das três edições do DARPA são apresentados na Figura 1.1.



Figura 1.1 - Veículos vencedores das três edições do DARPA. StandStorm (edição 2004) à esquerda, Stanley (edição 2005) no centro e Boss (edição 2007) à direita.

Em todos os projetos de veículos autônomos, cinco componentes funcionais básicos são envolvidos, como descritos por Whyte (2001), são eles: planejamento, navegação, mobilidade, localização e comunicação. Esses componentes se interagem como mostrado no diagrama da Figura 1.2.

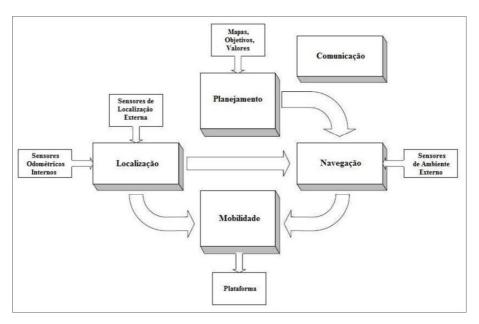


Figura 1.2 - Diagrama de relacionamento entre os diferentes componentes funcionais de um veículo autônomo (traduzido de Whyte, 2001)

A função da **navegação** é capturar as informações do ambiente externo e criar uma representação interna do ambiente que pode ser posteriormente utilizado para tomada de decisão.

A **mobilidade** está relacionada aos elementos físicos do veículo, a iteração do veículo com o terreno e o efeito do controle do veículo sobre o terreno. Mobilidade pode ser entendida como o executor do sistema global e o resultado observável do sistema como um todo.

A componente **localização** fornece estimativas de posicionamento, atitude, velocidade e aceleração do veículo.

A **comunicação** fornece a ligação entre o veículo e qualquer outro elemento do sistema, incluindo outros veículos e o operador.

O **planejamento** também pode ser chamado de missão ou planejamento de tarefas, é responsável por gerar rotas para o sistema como um todo. Esse componente não tem relacionamento direto com qualquer sinal de entrada dos sensores nem com o sinal de saída do controlador. No entanto, ele

deve usar uma compreensão desses em conjunto com mapas e com os objetivos da missão definida, para produzir comandos de navegação apropriados.

1.2.1. Pesquisa sobre veículos autônomos dentro do grupo GIGA

A linha de pesquisa sobre desenvolvimento de tecnologias ligada ao tema veículo autônomo se iniciou, dentro do grupo GIGA, com o trabalho de Gonçalves (2011) através da adaptação de um veículo rádio controlado denominado Colossus (Figura 1.3), fabricado e comercializado no Brasil entre 1984 e 1990 pela empresa Brinquedos Estrela S/A. O autor adicionou diversas placas de circuitos eletrônicos, sensores e bateria ao veículo e desenvolveu um algoritmo de controle do motor e direção das rodas.

A tarefa do veículo foi visitar pontos estabelecidos previamente utilizando apenas informações de posicionamento adquiridas via *GPS* (*Global Position system*) de navegação acoplado. Além disso, o veículo possui sensor de distância, implantado na parte frontal da plataforma, que fornecem dados para o segmento de navegação e impeça colisões com obstáculos.



Figura 1.3 - Veículo autônomo desenvolvido por Gonçalves (2011)

Em 2010, o grupo conseguiu recursos, através de proposta submetida ao edital CT transportes do CNPq, para adquirir duas novas plataformas (Figura 1.4). Com esses novos recursos o grupo espera desenvolver aplicações que utilizam, além das informações captadas pelos sensores do veículo, informações recebidas de outras plataformas para auxiliar na tomada de decisão ao logo da execução do percurso (como obstáculos nas vias ou falha de algum dos veículos envolvidos na

missão), isso possibilitará que os veículos alterem suas rotas de maneira dinâmica para garantir que todos os pontos envolvidos na missão sejam visitados.



Figura 1.4 - Plataforma adquirida pelo grupo GIGA para a pesquisa de veículos autônomos

Outro trabalho está sendo finalizado por Anderson Mori, integrante do GIGA, focado no segmento de localização, com objetivo de integrar ao sistema informações adquiridas por sensores inerciais de forma a melhorar a localização dos veículos em lugares sem sinal GNSS.

Esse trabalho está inserido no segmento de planejamento e tem o intuito principal de produzir rotas otimizadas para um conjunto de veículos, considerando os pontos a serem visitados e as janelas de tempo relacionadas a eles. A metodologia adotada para construir o algoritmo de roteirização foi baseada em metaheurística, como será descrito na próxima seção.

1.3. Estrutura do algoritmo

O algoritmo desenvolvido para o VRPTW foi baseado na metaheurística algoritmo genético onde os cromossomos representaram a ordem de atendimento dos clientes pelos veículos e a estrutura de codificação dessa informação não utilizou delimitadores de rota. Para realizar a divisão ótima de cada cromossomo em rotas, foi utilizado o procedimento split de Prins (2004) adaptado para o VRPTW, uma vez que o procedimento foi criado originalmente para tratar o VRP. A população inicial foi formada por uma parte construída com cromossomos criados aleatoriamente e outra parte

construída através da heurística de inserção I1 de Solomon (1987), com quatro formas diferentes de inserir o primeiro cliente em cada rota:

- Cliente mais distante do depósito.
- Cliente que apresenta menor instante final da janela de tempo.
- Cliente escolhido através da equação ponderada de Tan et al. (2001).
- Cliente inicial aleatório.

Na fase de recombinação, foram utilizados quatro tipos de crossover: uniforme, 2 pontos, heurístico e *PMX* (*Partially Mapped Crossover*); e um operador de mutação baseado em busca gulosa. A cada geração foram aplicados princípios de elitismo e pós-otimização, sendo que para realizar a pós-otimização foi utilizado a heurística λ-*Interchange* de Osman (1993). O algoritmo foi testado nos conjuntos C1, C2, R1, R2, RC1 e RC2 de Solomon (1987) e os resultados foram comparados com os melhores resultados encontrados na literatura.

1.4. Delineamento do Trabalho

O capítulo 2 apresenta uma taxonomia de classificação de problemas de roteirização de veículos proposta por Eksioglu *et al.* (2009).Uma revisão da literatura com abordagens de solução para problemas de roteirização, dividida em três grupos: métodos exatos, heurísticas e metaheurísticas. Nesta seção também são apresentados os melhores resultados da literatura para o *VRPTW*, considerando métodos exatos e heurísticos.

O capítulo 3 apresenta o modelo matemático e as características do problema clássico de roteirização de veículos e sua variação incluindo janelas de tempo.

O método utilizado nessa pesquisa para resolver o problema de roteirização de veículos com janelas de tempo é descrito no capítulo 4.

O capítulo 5 relata os testes realizados para calibrar os parâmetros utilizados e apresenta uma comparação do resultado obtido pelo algoritmo proposto com os melhores resultados publicados na literatura para as instâncias de Solomon (1987).

Finalmente, no capítulo 6 são apresentadas as conclusões e considerações desse trabalho.

2. REVISÃO BIBLIOGRÁFICA

O problema de roteirização de veículos com janelas de tempo tem sido estudado extensivamente com diferentes técnicas de solução. Os principais métodos exatos utilizados para resolver o problema são: *branch-and-bound*, *branch-and-cut* e *branch-and-price*. Todos eles utilizam alguma técnica de relaxação do problema de programação inteira, combinada, dependendo do método, com o uso de planos de cortes e/ou de geração de colunas.

As principais heurísticas pesquisadas para o problema são algoritmos de construção e refinamento. Esses algoritmos são capazes de explorar apenas uma parte do espaço de soluções fornecendo bons resultados com baixo tempo de processamento.

As metaherurísticas são técnicas mais avançadas que as heurísticas capazes de explorar uma parte maior do espaço de solução e sair de ótimos locais. As metaheurísticas são responsáveis pelos melhores resultados na literatura. Segundo Cunha (2006) as metaheurísticas podem ser definidas como as estratégias e técnicas mais recentes e avançadas, que guiam outras heurísticas a fim de encontrar soluções melhores, ultrapassando o ponto de parada das heurísticas tradicionais. As principais metaheurísticas encontradas na literatura relacionadas ao problema de roteirização de veículos com janelas de tempo são: busca tabu, *simulated annealing*, algoritmo genético e GRASP (*Greedy Randomized Adaptive Search Procedure*).

A seguir são listadas as principais contribuições da literatura para o problema de roteirização de veículos e suas extensões.

2.1. Taxonomia dos problemas de roteirização de veículos

Os problemas de roteirização de veículos tem se destacado como um dos principais problemas da cadeia logística. Para empresas que dependem do transporte rodoviário, a eficiente operação de seus veículos é fundamental para a aumentar a competitividade e o valor agregado de seus produtos, consequência disso é o grande número de trabalhos publicados nessa área. Eksioglu *et al.* (2009) apresentam um estudo estatístico dos trabalhos publicados na literatura sobre *VRP* de 1955 a 2005, com total de 1494 publicações e uma taxa de publicações com crescimento anual de 6,09%.

Desde os primeiros estudos, realizados na década de 50 do século XX, como Dantzig e Ramser (1959), muitas variações do problema são propostas de modo a incorporar as características dos problemas encontrados em complexas redes de distribuição. Tais características estão relacionadas com diferentes tipos de restrições, com a função objetivo, com as variáveis de decisão e com as hipóteses (Belfiore, 2006).

Tendo em vista o grande número de variações que o problema clássico de roteirização de veículos pode sofrer, alguns autores propuseram taxonomias para classificar tais problemas dentro de uma abordagem sistemática, tais como: Christofides (1985), Bodin e Golden (1981), Bodin *et al.* (1983), Assad (1988) e Ronen (1988). A taxonomia mais recente e completa foi publicada por Eksioglu *et al.* (2009), como mostra a Tabela 2.1.

Tabela 2.1 - Taxonomia do problema de roteirização de veículos segundo Eksioglu et al. (2009)

	Teoria		
		Métodos exatos	
	3.4% 1 1' 1	Heurísticas	
Tipo de estudo	e estudo Métodos aplicados	Simulação	
-		Métodos de solução em tempo real	
	Implementação documentada		
	Levantamento, revisão ou metapesquisa		
	Número de paradas na rota	Conhecido (determinístico)	
		Parcialmente conhecido, parcialmente	
		probabilístico Fracionamento permitido	
	Restrições de carregamento fracionado	*	
	, ,	Fracionamento não permitido Determinística	
		Estocástica	
	Tipo da demanda do cliente	Desconhecida Desconhecida	
		Determinística	
	Quantidade de requisições de novos clientes	Estocástica	
		Desconhecida	
	Tempo de serviço e de espera	Determinístico	
		Dependente do tempo	
Características		Dependente do tipo de veículo	
do cenário	Tempo de serviço e de espera	Estocástico	
		Desconhecido	
		Janela de tempo leve	
	Estrutura da janela de tempo	Janela de tempo restrita	
	T.	Ambas	
		Período simples	
	Horizonte de tempo	Períodos múltiplos	
	Backhauls	Requisição de coletas e entregas simultâneas	
		dos nós	
		Requisição de linehaul ou backhaul não	
		simultâneas dos nós	
		Precedência e restrições de agrupamento	
	Restrições nos arcos ou nos nós	Restrições de subconjunto	
		Permitido refazer rota	

		Rede direcionada
	Esquema da rede de transporte	Rede não direcionada
		Clientes nos nós
	Endereços das localidades (clientes)	Instâncias do arco de roteamento
		Urbana (dispersa com padrão)
	Localização geográfica dos clientes	Rural (dispersa aleatoriamente)
	Localização geografica dos effentes	Combinado
		Única origem
	Número de origens	Múltiplas origens
	Número de instalações de como comento e	Único depósito
	Número de instalações de carregamento e descarregamento (depósito)	Múltiplos depósitos
	deseuregamento (deposito)	Restrições nos clientes
		Restrições nas vias
	Tipo de janela de tempo	Restrições nos depósitos/hubs
		Restrições nos motoristas/veículos
		Exatamente <i>n</i> veículos
Comontanísticos	Número de veículos	Até n veículos
Características físicas do	Numero de verculos	Ilimitado
problema		
problema	Consideração de capacidade	Veículos capacitados
		Veículos não capacitados
		Veículos similares
	Homogonoido do dos vistavlos (comocidado)	Veículos carregam tipos específicos de carga
	Homogeneidade dos veículos (capacidade)	Veículos heterogêneos
		Clientes devem ser visitados por um tipo específico de veículo
		Determinístico
	Tempo de viagem	Função do horário atual
		Estocástico
		Desconhecido
		Dependente do tempo de viagem
		Dependente da distância
	Custo de transporte	Dependente do veículo
		Dependente da operação
		Função do atraso
		Relacionado ao risco/perigo
	E	Estática
	Evolução da informação	Parcialmente dinâmica
		Determinístico
		Estocástico
Características	Qualidade da informação	Previsão
da informação		Desconhecido (tempo real)
		Local
	Disponibilidade da informação	Global
	I	Centralizado
		Descentralizado
		Dados do mundo real
Características		Dados sintéticos
dos dados		Combinado (real e sintético)
	Não utilizam dados	(

Diante da grande diversidade de fatores e condicionantes que aparecem em problemas de roteirização, como mostrado na Tabela 1, para que possam ser atingidas soluções de qualidade e para que sejam suprimidas as demandas particulares de cada problema originado de situações reais,

são necessárias estratégias matemáticas eficazes e robustas o suficiente para serem aplicadas nos mais diferentes problemas (Araújo, 2008).

2.1. Métodos Exatos

Achuthan *et al.* (2003) resolveram um problema de roteirização de veículos considerando frota homogênea com número fixo e variável de veículos e restrição de capacidade de veículos, através de novos algoritmos de plano de corte, que foram implementados em um algoritmo *branch-and-cut*.

Baker (1982) utilizou a estratégia *branch-and-bound* para um problema de roteirização de veículo com janela de tempo com apenas um veículo capacitado. O objetivo do modelo é minimizar o tempo total das rotas.

Bard *et al.* (2002) utilizaram a técnica *branch-and-cut* para o problema de roteirização de veículos com janelas de tempo com frota homogênea. O objetivo é minimizar a distância total percorrida e o número de veículos necessários.

Chabrier (2005) propôs melhorias para o método de decomposição Dantzig-Wolfe e geração de colunas para o problema de roteirização de veículos com janelas de tempo. O autor apresentou limites inferiores de melhor qualidade obtidos com o problema mestre restrito relaxado e técnicas para reduzir o impacto na eficiência de levar a restrição de ciclo em conta.

Christofides *et al.* (1981) desenvolveram algoritmos baseados em métodos exatos – relaxação lagrangeana e programação dinâmica relaxada dos problemas da árvore de cobertura mínima e do caminho mínimo – para o problema clássico de roteirização de veículos. O problema tem como características: frota homogênea, restrição de capacidade máxima dos veículos e duração máxima de jornada de trabalho.

Cook e Rich (1999) apresentaram uma nova aplicação do trabalho de Kohl *et al.* (1999) utilizando o algoritmo de mínimo corte randômico de Karger e Stein (1993) para gerar os planos de corte.

Desrochers *et al.* (1992) utilizaram geração de colunas para solução do problema de roteirização de veículos com janelas de tempo e frota homogênea. Os autores aplicaram o método ao conjunto de

problemas R1, C1 e RC1 de Solomon (1987) com 100 clientes e em problemas derivados desses com 50 e 25 clientes. A solução ótima foi encontrada em todos os 29 problemas com 25 clientes, em 14 problemas com 50 clientes e em 7 problemas com 100 clientes. O método se mostrou bastante eficiente em problemas com clientes agrupados (grupo C1 de problemas) resolvendo 21 de 27 problemas, sendo que 5 problemas envolviam 100 clientes. Por outro lado, o método foi menos eficiente no grupo de problemas RC1 que mistura clientes agrupados e dispersos aleatoriamente resolvendo apenas 8 problemas com 25 clientes.

Irnich e Villeneuve (2005) apresentaram um algoritmo *branch-and-price* com eliminação de k-ciclos ($k \ge 3$) na geração de colunas para o problema de roteirização de veículos com janelas de tempo.

Kallehauge *et al.* (2000) propuseram uma abordagem baseada em relaxação lagrangeana para resolver o problema de roteirização de veículos com janelas de tempo. O algoritmo de plano de corte região de confiança foi combinado com um algoritmo Dantzig-Wolfe para encontrar soluções inteiras em um esquema *branch-and-bound*. O nó raiz da árvore *branch-and-bound* é solucionado por um algoritmo de plano de corte e, se uma solução inteira não é obtida, um algoritmo Dantzig-Wolfe passa ser utilizado na árvore de nós. O algoritmo teve sucesso ao resolver problemas ainda não solucionados nas instâncias de Solomon (1987).

Kohl *et al.* (1999) apresentaram uma proposta baseada em um método de plano de corte de duas fases denominado 2-*path cuts*, seguido de uma busca *branch-and-bound* como geração de colunas. Os autores utilizam um eficiente algoritmo de separação, incorporado no problema mestre da decomposição Dantzig-Wolfe, para encontrar tais pontos de cortes. O subproblema é um problema de caminho mínimo com restrições de capacidade e janelas de tempo. O algoritmo foi testado nas instâncias de Solomon (1987) e conseguiu atingir soluções ótimas em 14 dos 56 problemas envolvendo 100 clientes.

Kolen *et al.* (1987) desenvolveram um algoritmo *branch-and-bound* para o problema de roteirização de veículos com janelas de tempo. O problema considera frota homogênea e capacitada. O objetivo é minimizar a distância total percorrida.

Larsen (1999) desenvolveu um algoritmo baseado na decomposição Dantzig-Wolfe onde o problema mestre é um problema particionado relaxado que garante que cada cliente é visitado apenas uma vez, enquanto o subproblema é um problema de caminho mínimo com restrições de capacidade e janelas de tempo. O problema mestre é utilizado para reduzir custos que são

calculados em cada arco. Esses custos são utilizados no subproblema para garantir rotas que tenham origem e destino no depósito.

2.2. Heurísticas

Antes e Derigs (1995) resolveram o problema de roteirização de veículos com janelas de tempo através de um algoritmo de construção e melhoria que é capaz de lidar com muitas rotas simultaneamente. A abordagem é baseada no conceito de negociação entre clientes e rotas onde o preço é calculado utilizando a heurística II de Solomon (1987). Os autores propuseram também um procedimento de pós-otimização, onde os clientes mais ineficientes são removidos das rotas e em seguida reinseridos utilizando um procedimento de negociação.

Clarke e Wright (1964) desenvolveram uma heurística de economias que têm sido muito utilizadas em problemas de roteirização de veículos. O método permite incorporar diversos tipos de restrições, como a inclusão de janelas de tempo.

Cunha e Gualda (1997) apresentam três heurísticas baseadas na relaxação lagrangeana para o problema de roteirização de veículos com janelas de tempo considerando frota heterogênea. As heurísticas apresentaram resultados equivalentes ou superiores em comparação aos modelos testados na literatura, sendo que a heurística de agrupamento e alocação sequencial foi superior em relação às outras duas.

Frizzell e Giffin (1995) apresentaram a formulação matemática para o problema de roteirização de veículos com janelas de tempo e entregas fracionadas, considerando frota homogênea, demanda dos clientes menor que a capacidade dos veículos e restrições de janelas de tempo. Os autores propuseram uma heurística construtiva baseada na urgência dos clientes e duas heurísticas de melhoria que foram aplicadas a diversos problemas encontrados na literatura.

Gillett e Miller (1974) estudaram a resolução do problema clássico de roteirização de veículos utilizando uma técnica de varredura (também conhecida como *sweep*) que basicamente consiste em agrupar os clientes segundo uma determinada regra de varredura e depois construir rotas econômicas para cada agrupamento.

Ioannou *et al.* (2001) utilizaram um novo critério de seleção e inserção de clientes baseados na minimização da função gulosa desenvolvida por Atkinson (1994), onde o cliente escolhido para ser inserido na rota deve minimizar o impacto nos clientes já inseridos na rota, nos que ainda não foram inseridos e na janela de tempo do cliente que está sendo escolhido para fazer parte da rota.

Potvin e Rosseau (1993) apresentaram uma versão adaptada da heurística de inserção sequencial de Solomon (1987), em que as rotas são construídas paralelamente tanto no critério de início das rotas quanto de inserção. O algoritmo foi aplicado ao problema de roteirização de veículos com janelas de tempo utilizando frota homogênea.

Rochat e Taillard (1995) apresentaram uma técnica probabilística para diversificar, intensificar e paralelizar uma busca local adaptada para problemas de roteirização de veículos. Os autores ainda propuseram um procedimento de pós-otimização que permite melhorar a solução com pouco aumento de tempo computacional. O algoritmo foi testado em instâncias do problema de roteirização de veículos e sua variação com janelas de tempo obtendo resultados satisfatórios nos dois problemas.

Russell (1995) utilizou heurística de inserção paralela e algoritmo de troca de nós entre rotas para resolver o problema de roteirização de veículos com janelas de tempo com frota homogênea. O objetivo foi minimizar o número de veículos, o tempo total de rota e a distância total percorrida.

Shaw (1998) utilizou um método de busca local denominado *Large Neighbourhood Search* (LNS) para solucionar o problema de roteirização de veículos com janelas de tempo. Essa técnica explora grandes vizinhanças da solução corrente, selecionando um número de clientes visitados para remover do plano de rota e reinserir utilizando uma árvore de busca baseada nas restrições. O autor afirma que o método é competitivo com as metaheurísticas tomando como base os resultados obtidos em seus testes.

Solomon (1987) apresentou um conjunto de heurísticas de construção baseadas em inserção sequencial para o problema de roteirização de veículos com janelas de tempo. Essas heurísticas se tornaram base para outros autores pelo bom resultado das soluções e baixo custo computacional. O problema considera a frota de veículos homogênea e veículos com capacidade limitada.

Thangiah *et al.* (1994) descreveram um algoritmo de duas fases para o problema de roteirização de veículos com janelas de tempo. Na primeira fase, uma população inicial foi gerada utilizando a heurística *Push Forward Insertion* de Solomon (1987) e uma heurística baseada na abordagem

cluster-first route-second denominada Heurística de Setoramento Genético (Genetic Sectoring Heuristic) de Thangiah (1993). Na segunda fase um mecanismo denominado λ -interchange é utilizado para mover clientes entre rotas, gerando uma vizinhança de soluções para o problema.

Thompson e Psaraftis (1993) apresentaram um método de melhoria baseado no conceito de transferência cíclica (*cycle transfer*) para problemas de roteirização de veículos com frota homogênea.

Uma boa fonte de consulta de algumas das técnicas descritas acima se encontra em Bräysy e Gendreau (2005). Nesse artigo, os autores descrevem as principais técnicas de solução do problema de roteirização de veículos com janelas de tempo e compara os resultados em termos de quantidades de veículos utilizados e distância total percorrida.

2.3. Metaheurísticas

Os próximos itens apresentam uma breve descrição do princípio de cada método metaheurístico utilizado para resolver problemas de roteirização de veículos. Alguns trabalhos que utilizaram esses métodos são também apresentados em seguida, onde são destacadas as principais características de aplicação.

Busca Tabu

A metaheurística busca tabu foi introduzida inicialmente por Glover (1989) e Fisher *et al.* (1997), e consiste em uma busca em vizinhança que procura não percorrer regiões do espaço de soluções já visitadas e que não sejam promissoras. Para isso, a estratégia utiliza uma lista de movimentos proibidos denominados de "tabu". Tais movimentos, quando realizados, podem levar a busca às regiões menos promissoras, descritas anteriormente.

No entanto, a lista tabu recebe um fator temporal em que os movimentos proibidos no momento corrente podem ser permitidos em iterações futuras. O conjunto de elementos proibidos da busca tabu faz parte de uma memória evolutiva, o que possibilita sua alteração de acordo com o tempo e circunstância (Cunha, 2006).

A busca parte de uma solução inicial e explora sua vizinhança através de trocas de clientes entre rotas ou realocação de um cliente dentro de sua própria rota. Cada movimento gera uma nova solução e a cada iteração um movimento de vizinhança é selecionado, sendo o processo repetido até se atingir um critério de parada (Cunha, 2006).

Aplicações utilizando busca tabu para resolução do problema clássico de roteirização de veículos foram abordadas por Barbarosoglu e Ozgur (1999), Gendreau *et al.* (1994), Kelly e Xu (1999) e Taillard (1993).

Badeau *et al.* (1997) desenvolveram uma proposta baseado na metaheurística busca tabu paralelo para um problema de roteirização de veículos com janelas de tempo rígidas e flexíveis, onde o objetivo é minimizar a distância total percorrida. A vizinhança foi baseada em movimentos de troca de arcos (*k-opt*), troca cruzada e trocas de clientes na mesma rota.

Cordeau *et al.* (2000) apresentam uma abordagem baseada na metaheurística busca tabu para o problema de roteirização de veículos com janelas de tempo e mais duas generalizações: o *VRP* periódico e o *VRP* com multi-depósitos. Segundo os autores, as principais características do algoritmo são a velocidade de processamento, a simplicidade e a flexibilidade.

Garcia *et al.* (1994) apresentaram uma implementação paralela da metaheurística busca tabu. A solução inicial do algoritmo foi construída com a heurística de inserção e testada no conjunto de problemas desenvolvido por Solomon (1987). O problema tratado possui custo total de roteirização envolvendo, além da distância total percorrida, o tempo total de espera.

Ho e Haugland (2004) implementaram a metaheurística busca tabu para o problema de roteirização de veículos com janelas de tempo e entregas fracionadas. Nesse problema, a frota foi considerada homogênea, porém a demanda dos clientes pode exceder a capacidade dos veículos. Além disso, os roteiros iniciam e terminam no depósito no horário máximo permitido.

Tan *et al.* (2001) exploraram diferentes algoritmos baseados em abordagens metaheurísticas como busca tabu, *simulated annealing* e algoritmo genético para o problema de roteirização de veículos com janelas de tempo. Todas as implementações foram aplicadas às instâncias de Solomon (1987) produzindo resultados competitivos.

Simulated annealing

O método *simulated annealing* recebe esse nome pelo fato de ser similar a um processo físico conhecido como recozimento (*annealing*) onde um material é aquecido até atingir o estado líquido e então é resfriado em seguida. Introduzida por Kirkpatrick *et al.* (1983), essa metaheurística se baseia no fato de que em altas temperaturas os átomos tem liberdade de se moverem livremente. À medida que a temperatura diminui, os átomos tendem a se cristalizar em um sólido. Para apresentar bons resultados, um resfriamento depende de uma temperatura inicial alta e de um resfriamento lento, a fim de evitar uma estrutura irregular e fraca, com alta energia em decorrência do esforço interno despendido (Cunha, 2000).

O algoritmo começa com uma solução inicial viável e, a partir dela, gera uma solução aleatória. Se a solução gerada é melhor que a primeira, ela passa a ser a solução atual, caso contrário ela só é aceita com certa probabilidade. Essa probabilidade de aceitação é determinada por uma temperatura que é gradualmente reduzida. Ao reduzir a temperatura, a seleção se torna cada vez mais seletiva na atribuição de uma nova solução como solução atual (Nasser e El-Sherbeny, 2010).

Bent e Hentenryck (2004) propuseram uma abordagem híbrida para o problema de roteirização de veículos com janelas de tempo utilizando *simulated annealing* e busca local em vizinhança de grande porte. O objetivo foi primeiramente minimizar o número de veículos utilizados e posteriormente a distância total percorrida.

Oliveira et al. (2007) implementaram um eficiente sistema híbrido que associa simulated annealing com a estratégia hill-climbing com reinício aleatório. A abordagem ocorreu inspirada na capacidade do simulated annealing em evoluir soluções para um dado problema e escapar de ótimos locais e na capacidade do hill-climbing de refinar soluções. Para completar o método, uma técnica chamada de reinício aleatório é aplicada a fim de lidar com a ideia de produzir soluções para variadas configurações do problema de roteirização de veículos com janelas de tempo, através de várias reinicializações do sistema.

Outro trabalho envolvendo simulated annealing foi apresentado por Osman (1993) que utilizou um algoritmo híbrido baseado nas metaheurísticas simulated annealing e busca tabu e um procedimento de melhoria denominado λ -interchange.

GRASP

O *GRASP* foi proposto por Feo e Resende (1989, 1995) é uma metaheurística iterativa que consiste em duas fases, a primeira, denominada fase de construção, é responsável por construir uma solução inicial factível e a segunda, denominada fase de busca local, consiste em explorar a vizinhança da solução inicial.

Na fase de construção, os possíveis movimentos de inserção são ranqueados de acordo com o incremento de custo que o elemento representa, medido através de uma função gulosa. Então um desses movimentos é escolhido aleatoriamente e o elemento relacionado a ele é escolhido para fazer parte da solução. Após isso, todos os custos referentes aos próximos movimentos são atualizados e então o próximo elemento é escolhido.

Na fase de busca local, a solução atual é melhorada até que o ótimo local seja obtido, nesta fase diversos procedimentos de melhoria podem ser empregados (Araújo, 2007).

Uma aplicação do *GRASP* foi proposta por Kontoravdis e Bard (1995) para o problema de roteirização de veículos com janelas de tempo com o objetivo de minimizar o número de veículos necessários e minimizar a distância total percorrida. O algoritmo foi testado no conjunto de problemas de Solomon (1987) e em problemas reais com até 417 clientes.

Algoritmos genéticos

Os algoritmos genéticos são procedimentos probabilísticos de busca que se baseia na teoria da evolução genética. Diferentemente das outras metaheurísticas, os algoritmos genéticos trabalham com uma população de soluções que sofrem modificações a cada geração através da aplicação de operadores genéticos. A ideia central é que a solução escolhida seja aquela que tenha melhor adaptação relativa, medida através de um valor de *fitness*, que envolve o objetivo global da otimização.

Alvarenga *et al.* (2007) aplicaram a metaheurística algoritmo genético a uma formulação particionada do problema de roteirização de veículos com janelas de tempo. O objetivo da abordagem foi encontrar rotas em soluções de mínimos locais e posteriormente verificar se são rotas ótimas, considerando a distância total percorrida da solução. Outro ponto importante do algoritmo são os oito diferentes tipos de mutação que aceleram a evolução dos indivíduos

rapidamente. O autor compara os resultados obtidos com as soluções publicadas na literatura e conclui que suas soluções são muito expressivas, estabelecendo novos valores de benchmark.

Baker e Ayechew (2003) implementaram um algoritmo genético para o problema de roteirização de veículos. O problema tem como características: frota homogênea, restrição de capacidade máxima dos veículos e duração máxima de jornada de trabalho. A população inicial foi criada utilizando dois métodos: o primeiro é baseado no trabalho de Gillet e Miller (1974) e o segundo no trabalho de Fisher e Jaikumar (1981). O algoritmo utilizou ainda rotinas de melhoria para acelerar a convergência da busca. Segundo os autores, este modelo é competitivo, em termos de tempo computacional e qualidade da solução, em relação a outras metaheurísticas para o problema de roteirização de veículos, como busca tabu e *simulated annealing*.

Cheng (2005) estudou um algoritmo genético híbrido, para resolver o problema de roteirização de veículos com janelas de tempo. Foi utilizada uma adaptação do trabalho de Prins (2004) para que o procedimento *split* passasse a considerar as janelas de tempo dos clientes no momento de dividir os cromossomos em rotas.

Ombuki *et al.* (2006) apresentaram um algoritmo genético multi-objetivo para o problema de roteirização com janelas de tempo. Os objetivos foram minimizar o número de veículos utilizados e a distância total percorrida. O algoritmo foi testado no conjunto de problemas de Solomon (1987).

Vieira (2008) desenvolveu um algoritmo genético para o problema de roteirização de veículos com janelas de tempo. O trabalho apresenta diversos tipos de cruzamentos e mutações, além de técnicas avançadas como *hill-climbing*.

Prins (2004) apresentou um algoritmo genético híbrido para o problema clássico de roteirização de veículos utilizando codificação de cromossomo sem delimitadores de rotas, o que permitiu a utilização do operador *OX* (*order crossover*), que, segundo o autor, é o método de cruzamento que obtém maior sucesso nos algoritmos genéticos aplicados a problemas do tipo caixeiro-viajante e de roteirização de veículos. O autor ainda propõe um procedimento de particionamento do roteiro gigante, denominado procedimento s*plit* (*split procedure*) que divide o cromossomo em rotas de maneira ótima. O algoritmo foi testado nas instâncias de Christofides *et al.* (1979) e Golden *et al.* (1998) obtendo excelentes resultados.

Potvin e Bengio (1996) implementaram uma proposta utilizando o algoritmo genético para o problema de roteirização de veículos com janelas de tempo. A população inicial foi gerada através

da heurística de inserção I1 de Solomon (1987) e testada no conjunto de problemas do mesmo autor.

Tan *et al.* (2006) apresentaram um algoritmo multi-objetivo evolucionário híbrido com operadores genéticos especializados. Os autores utilizaram operadores genéticos especializados e representação do cromossomo de comprimento variável para acomodar a codificação dos clientes em sequência no problema de roteirização de veículos com janelas de tempo.

Ursani et al. (2011) introduziram uma abordagem denominada Estrutura de Otimização Localizada (LOF, Localized Optimization Framework) para o problema de roteirização de veículos com janelas de tempo. O algoritmo, denominando Algoritmo Genético Localizado (LGA, Localized Genetic Algorithm), decompõe o problema original e o resolve de forma iterativa em duas fases, Optimization e De-Optimization. A primeira, baseada na metaheurística algoritmo genético, é utilizada para resolver partes do problema enquanto a segunda é aplicada para o problema todo. O trabalho apresenta bons resultados quando comparados as melhores soluções publicadas.

2.4. Melhores resultados da literatura para o *VRPTW*

A principal forma de avaliar a eficiência e eficácia dos diferentes métodos de resolução do problema de roteirização de veículos com janelas de tempo é através da comparação dos resultados obtidos, quando aplicado a uma base de problemas comum, dado um objetivo pré-definido. Atualmente, a principal base de comparação são as instâncias criadas por Solomon (1987) composta por 56 problemas envolvendo 100 clientes (mais detalhes são apresentados na seção 5.1).

A Tabela 2.2 apresenta os principais resultados publicados na literatura relacionados às instâncias de Solomon (1987). Atualmente, 37 dos 56 problemas foram resolvidos de forma ótima, sendo que os 17 problemas que permanecem sem solução exata servem como estimulo para que os pesquisadores desenvolvam abordagens cada vez mais robustas. Por outro lado, os métodos heurísticos são cada vez mais explorados, seja através do aperfeiçoamento de heurísticas existentes, tornando-as mais eficazes, ou através de pesquisas envolvendo métodos baseados em metaheurísticas, fornecendo resultados muito satisfatórios em diversas instâncias.

 ${\bf Tabela~2.2~-Levantamento~dos~melhores~resultados~publicados~na~literatura~envolvendo~m\'etodos~exatos~e~heur\'esticos,~aplicados~ao~\it VRPTW}$

Instância	Métodos exatos		Autores	Métodos heurísticos		Autores
	NV	DT		NV	DT	
C101	10	827,3	Kohl et al. (1999)	10	828,940	Rochat e Taillard (1995)
C102	10	827,3	Kohl et al. (1999)	10	828,940	Rochat e Taillard (1995)
C103	10	826,3	Kohl et al. (1999)	10	828,060	Rochat e Taillard (1995)
C104	10	822,9	Kohl et al. (1999)	10	824,780	Rochat e Taillard (1995)
C105	10	827,3	Kohl et al. (1999)	10	828,940	Rochat e Taillard (1995)
C106	10	827,3	Kohl et al. (1999)	10	828,940	Rochat e Taillard (1995)
C107	10	827,3	Kohl et al. (1999)	10	828,940	Rochat e Taillard (1995)
C108	10	827,3	Kohl et al. (1999)	10	828,940	Rochat e Taillard (1995)
C109	10	827,3	Kohl et al. (1999)	10	828,940	Rochat e Taillard (1995)
C201	3	589,1	Cook e Rich (1999) + Kallehauge <i>et al.</i> (2000)	3	591,560	Rochat e Taillard (1995)
C202	3	589,1	Cook e Rich (1999) + Kallehauge <i>et al.</i> (2000)	3	591,560	Rochat e Taillard (1995)
C203	3	588,7	Kallehauge et al. (2000)	3	591,170	Rochat e Taillard (1995)
C204	3	588,1	Irnich e Villeneuve (2005)	3	590,600	Rochat e Taillard (1995)
C205	3	586,4	Cook e Rich (1999) + Kallehauge <i>et al.</i> (2000)	3	588,880	Rochat e Taillard (1995)
C206	3	586,0	Cook e Rich (1999) + Kallehauge <i>et al.</i> (2000)	3	588,490	Rochat e Taillard (1995)
C207	3	585,8	Cook e Rich (1999) + Kallehauge <i>et al.</i> (2000)	3	588,290	Rochat e Taillard (1995)
C208	3	585,8	Kallehauge et al. (2000)	3	588,320	Rochat e Taillard (1995)
R101	20	1637,7	Kohl et al. (1999)	20	1642,870	Alvarenga et al. (2007)
R102	18	1466,6	Kohl et al. (1999)	18	1472,620	Alvarenga et al. (2007)
R103	14	1208,7	Cook e Rich (1999) + Larsen (1999)	14	1213,620	Rochat e Taillard (1995)
R104	11	971,5	Irnich e Villeneuve (2005)	10	982,010	Rochat e Taillard (1995)
R105	15	1355,3	Kohl et al. (1999)	15	1360,783	Alvarenga et al. (2007)
R106	13	1234,6	Cook e Rich (1999) + Kallehauge <i>et al.</i> (2000)	13	1241,518	Alvarenga et al. (2007)
R107	11	1064,6	Cook e Rich (1999) + Kallehauge <i>et al.</i> (2000)	11	1076,125	Alvarenga et al. (2007)
R108				10	948,573	Alvarenga et al. (2007)
R109	13	1146,9	Cook e Rich (1999) + Kallehauge <i>et al.</i> (2000)	13	1151,839	Alvarenga et al. (2007)
R110	12	1068,0	Cook e Rich (1999) + Kallehauge <i>et al.</i> (2000)	11	1080,360	Rochat e Taillard (1995)
R111	12	1048,7	Cook e Rich (1999) + Kallehauge <i>et al.</i> (2000)	12	1053,496	Alvarenga et al. (2007)
R112				10	953,630	Rochat e Taillard (1995)
R201	8	1143,2	Kallehauge et al. (2000)	9	1147,800	Oliveira et al. (2007)
R202				5	1039,320	Oliveira et al. (2007)
R203				5	874,870	Oliveira et al. (2007)
R204				3	735,800	Oliveira et al. (2007)
R205				5	954,160	Oliveira et al. (2007)
R206				4	884,250	Oliveira et al. (2007)
R207				4	797,990	Oliveira et al. (2007)
R208				3	705,620	Oliveira et al. (2007)
R209				5	860,110	Alvarenga et al. (2007)
R210				5	910,980	Oliveira et al. (2007)
R211				4	755,820	Oliveira et al. (2007)
RC101	15	1619,8	Kohl et al. (1999)	15	1623,580	Rochat e Taillard (1995)
RC102	14	1457,4	Cook e Rich (1999) + Kallehauge <i>et al.</i> (2000)	14	1466,840	Alvarenga et al. (2007)
RC103	11	1258,0	Cook e Rich (1999) + Kallehauge <i>et al.</i> (2000)	11	1261,670	Shaw (1998)

RC104				10	1135,480	Cordeau et al. (2000)
RC105	15	1513,7	Kohl et al. (1999)	16	1518,600	Alvarenga et al. (2007)
RC106				13	1377,352	Alvarenga et al. (2007)
RC107	12	1207,8	Irnich e Villeneuve (2005)	12	1212,830	Alvarenga et al. (2007)
RC108	11	1114,2	Irnich e Villeneuve (2005)	11	1117,526	Alvarenga et al. (2007)
RC201	9	1261,8	Kallehauge et al. (2000)	9	1266,110	Oliveira et al. (2007)
RC202	8	1092,3	Irnich e Villeneuve (2005) + Chabrier (2005)	8	1096,750	Oliveira et al. (2007)
RC203				5	926,890	Oliveira et al. (2007)
RC204				4	786,380	Oliveira et al. (2007)
RC205	7	1154,0	Irnich e Villeneuve (2005) + Chabrier (2005)	7	1157,550	Oliveira et al. (2007)
RC206				7	1056,210	Oliveira et al. (2007)
RC207				7	966,080	Oliveira et al. (2007)
RC208				4	779,840	Alvarenga et al. (2007)

Um ponto importante que diferencia o *benchmark* das soluções exatas é a precisão, de apenas um dígito, no cálculo da distância, enquanto as soluções heurísticas utilizam até três dígitos. Essa diferença de precisão resulta em diferentes distâncias e acabam confundindo alguns pesquisadores que comparam seus resultados, decorrentes de abordagens heurísticas, com as distâncias originadas das soluções utilizando métodos exatos. Ursani *et al.* (2011) produziram resultados utilizando as duas precisões no cálculo das distâncias, viabilizando a comparação com os resultados dos métodos exatos e heurísticos. Essa pesquisa também adotou essa forma de comparação nos testes computacionais.

Grande parte da proposta de solução desenvolvida nessa pesquisa encontrou inspiração em trabalhos publicados na literatura, alguns citados nessa revisão bibliográfica, e adaptados conforme necessário. Na construção da população inicial, a heurística *Push Forward Heuristic* desenvolvida por Solomon (1987), e posteriormente por diversos autores como Antes e Derigs (1995), Potvin e Rosseau (1993), Thangiah *et al.* (1994), Garcia *et al.* (1994), Alvarenga *et al.* (2007) e Potvin e Bengio (1996), foi utilizada com quatro regras diferentes para escolher o primeiro cliente a ser inserido na rota, sendo que em uma delas a escolha é aleatória, duas delas foram aproveitadas do trabalho de Solomon (1987) e a última foi baseada no trabalho de Thangiah *et al.* (1994).

O trabalho de Prins (2004) contribuiu para a escolha da representação do cromossomo e na forma de dividi-lo em rotas. Nesse ponto, um estudo da adaptação desse procedimento, descrito em Cheng (2005), para considerar as janelas de tempo foi necessário.

Quatro métodos de *crossover* e um de mutação foram implementados nessa pesquisa e estão baseados no trabalho de Linden (2006), Goldberg e Lingle (1985) e Vieira (2008). O procedimento

de elitismo foi inspirado em Linden (2006) e o método λ -interchange utilizado na pós-otimização foi baseado no trabalho de Thangiah (1995).

A formulação apresentada em Bodin *et al.* (1983) para o problema clássico de roteirização de veículos foi estudada e aproveitada nessa pesquisa para contextualizar o problema. A variação, considerando janelas de tempo, foi encontrada em Larsen (1999) e Ombuki *et al.* (2006). A formulação matemática e as caraterísticas dos dois problemas são apresentadas na próxima seção.

3. CARACTERIZAÇÃO DO PROBLEMA

Este capítulo apresenta o modelo clássico do problema de roteirização de veículos e o modelo do problema de roteirização de veículos com janelas de tempo que é uma variação do primeiro. Além disso, será apresentada a atual taxonomia de classificação de problemas de roteirização.

3.1. O problema clássico de roteirização de veículos

O problema clássico de roteirização de veículos consiste em definir rotas de entrega para uma frota homogênea de veículos que atendam a um conjunto de clientes a partir de um depósito de origem. O objetivo é encontrar um conjunto de rotas que minimize a distância total percorrida e que atendam as demandas de todos os clientes. O problema assegura que as rotas devem iniciar e terminar no depósito e que todos os clientes (demanda determinística) devem ser atendidos uma única vez, por um único veículo, sendo que esse possui capacidade limitada (Bodin *et al.* 1983).

Considerando a seguinte variável de decisão:

$$x_{ij}^{k} = \begin{cases} 1, \text{se o arco } (i,j) \text{ \'e percorrido pelo ve\'(culo } k \\ 0, \text{caso contr\'{a}rio} \end{cases}$$
 (1)

O modelo matemático proposto por Bodin *et al.* (1983) para o problema clássico de roteirização de veículos é dado por:

$$\min \sum_{i=1}^{N} \sum_{\substack{j=1\\j\neq i}}^{N} \sum_{k=1}^{K} c_{ij} x_{ij}^{k}$$
 (2)

sujeito a:
$$\sum_{k=1}^{K} \sum_{i=1}^{N} x_{ij}^{k} = 1$$
 $(j = 2, ..., n)$ (3)

$$\sum_{k=1}^{K} \sum_{i=1}^{N} x_{ij}^{k} = 1 \qquad (i = 2, ..., n)$$
 (4)

$$\sum_{i=1}^{N} x_{ih}^{k} - \sum_{j=1}^{N} x_{hj}^{k} = 0 \qquad (k = 1, ..., nv); \quad (h = 1, ..., n)$$
 (5)

$$\sum_{i=1}^{N} d_{i} \sum_{i=1}^{N} x_{ij}^{k} \leq C \qquad (k = 1, ..., nv)$$
 (6)

$$\sum_{j=2}^{N} x_{1j}^{k} \leq 1 \qquad (k = 1, ..., nv)$$
 (7)

$$\sum_{i=0}^{N} x_{1i}^{k} \leq 1 \qquad (k = 1, \dots, nv)$$
 (8)

$$x_{ij}^k \in \{0,1\}$$
 $(i,j=1,...,n); (k=1,...,nv)$ (9)

$$\sum_{i,j \in S} x_{ij}^{k} \leq |S| - 1 \qquad (\forall S \subseteq \{2, ..., n\}); \ (k = 1, ..., nv)$$
 (10)

Onde:

- N = conjunto de vértices $\{0, 1, ..., n, n+1\}$ em que o subconjunto $\{1, 2, ..., n\}$ representa os clientes e o subconjunto $\{0, n+1\}$ representa o depósito;
- $K = \text{conjunto dos veículos utilizados } \{1, ..., nv\};$
- C =capacidade máxima do veículo;
- S = qualquer subconjunto de vértices alocados a um veículo, excluindo-se o depósito, que não se repetem e que fazem parte de uma mesma rota;
- nv =último elemento do conjunto K de veículos;
- d_i = demanda do cliente i, e
- c_{ij} = custo de viagem do cliente i ao cliente j.

A equação (2) garante que a distância total percorrida seja minimizada. Alternativamente, para o VRP que considera frota heterogênea, o parâmetro c_{ij} deve ser alterado para c_{ij}^k passando a depender do tipo do veículo. As restrições (3) e (4) garantem que apenas um veículo fará a viagem entre dois pontos. (5) asseguram que se um veículo entra em um vértice, ele deve sair do mesmo vértice. As restrições (6) garantem que a capacidade máxima dos veículos seja atendida. As

restrições (7) e (8) garantem que cada veículo só sairá do depósito e retornará a ele uma única vez. As restrições (9) garantem que as variáveis de decisão sejam binárias. Finalmente, as restrições (10) asseguram que o número máximo de arcos dentro de cada rota não seja maior que o número de vértices de cada rota menos uma unidade, evitando dessa forma a ocorrência de rotas que não contenham o depósito, denominadas sub-rotas (subtours).

3.2. Problema de roteirização de veículos com janelas de tempo

O problema de roteirização de veículos com janelas de tempo (VRPTW) é uma generalização do problema clássico de roteirização de veículos (VRP) adicionando a restrição de janela de tempo para os clientes. Nesse problema, os veículos devem atender todos os clientes dentro do intervalo de tempo estipulado, sendo que os mesmos podem chegar antes do limite inferior da janela de tempo e esperar para começar o atendimento. Por outro lado, veículos atrasados não são permitidos, embora outras variações do problema permitam atraso dentro de uma abordagem envolvendo penalidades.

Os veículos são considerados homogêneos com uma capacidade fixa C. Nesse trabalho não foi estipulado um valor para a quantidade máxima de veículos. Dessa maneira, a frota é considerada ilimitada e o número de veículos escolhido na solução final será aquele que minimiza o custo de distribuição total.

Considerando a variável binária x_{ij}^k como no problema clássico VRP, a formulação matemática proposta por Larsen (1999) e Ombuki et al. (2006) para o VRPTW é dada por:

$$\min \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=1}^{K} c_{ij} x_{ij}^{k}$$
sujeito a:
$$\sum_{k=1}^{K} \sum_{j=1}^{N} x_{ij}^{k}$$

$$= 1 \qquad (i = 1, ..., n)$$

$$\sum_{i=1}^{N} d_{i} \sum_{j=0}^{N} x_{ij}^{k}$$

$$\leq C \qquad (k = 1, ..., nv)$$
(13)

sujeito a:
$$\sum_{k=1}^{K} \sum_{i=1}^{N} x_{ij}^{k}$$
 = 1 ($i = 1, ..., n$)

$$\sum_{i=1}^{N} d_i \sum_{i=0}^{N} x_{ij}^k \qquad \leq C \qquad (k = 1, \dots, nv)$$
 (13)

$$\sum_{j=0}^{N} x_{0j}^{k} = 1 \qquad (k = 1, ..., nv)$$
 (14)

$$\sum_{i=0}^{N} x_{ih}^{k} - \sum_{j=0}^{N} x_{hj}^{k} = 0 (k = 1, ..., nv); (h = 1, ..., n)$$

$$\sum_{i=0}^{N} x_{i,n+1}^{k} = 1 \qquad (k = 1, ..., nv)$$
 (16)

$$b_i^k + s_i + t_{ij} - M(1 - x_{ij}^k) \le b_j^k \qquad (i, j = 1, ..., n); \quad (k = 1, ..., nv)$$
 (17)

$$e_i \le b_i^k \le l_i$$
 $(i = 1, ..., n); (k = 1, ..., nv)$ (18)

$$x_{ij}^k \in \{0,1\}$$
 $(i, j = 1, ..., n); (k = 1, ..., nv)$ (19)

Onde:

- e_i = instante inicial da janela de tempo do cliente i;
- l_i = instante final da janela de tempo do cliente i;
- b_i^k = instante de início do atendimento do cliente i pelo veículo k;
- s_i = tempo de serviço do cliente i;
- t_{ij} = tempo de viagem do cliente i para o cliente j, e
- M =um número suficientemente grande.

A função objetivo (11) e as restrições (12) e (13) são as mesmas do problema clássico de roteirização de veículos. As equações (14), (15) e (16) são as restrições de fluxo que garantem que cada veículo deixe o depósito 0, passe pelos clientes e termine a rota no depósito n+1. As restrições (17) evitam que um veículo k, que esteja viajando do cliente i ao cliente j, não chegue ao seu destino antes do tempo de serviço no cliente i mais o tempo de viagem entre i e j ($s_i + t_{ij}$). Por fim, as restrições (18) garantem que o cliente i seja atendido dentro de sua janela de tempo.

Esse capítulo apresentou as principais características do *VRPTW* como uma generalização do *VRP* e sua formulação matemática. Como visto no capítulo 2, duas abordagens podem ser empregadas para solucionar o problema, métodos exatos e métodos heurísticos, sendo que nos métodos exatos a solução ótima é sempre garantida, ocorrendo muitas vezes em um tempo computacional elevado, por outro lado, os métodos heurísticos são abordagens que não garantem a solução ótima do problema, mas são muito eficientes em relação ao tempo de processamento.

O próximo capítulo apresenta a estratégia de solução do *VRPTW* utilizada nesse trabalho. A abordagem utilizada foi baseada na metaheurística algoritmo genético que se enquadra dentro do grupo de métodos heurísticos.

4. ESTRATÉGIA DE SOLUÇÃO

Nesse capítulo será detalhada a estratégia de solução proposta nessa pesquisa para o problema de roteirização de veículos com janelas de tempo. O algoritmo é baseado na metaheurística algoritmo genético utilizando cromossomo sem delimitadores de rotas. Para quebrar os cromossomos em rotas, foi utilizado um procedimento baseado no trabalho de Prins (2004) com adaptação para considerar as janelas de tempo dos clientes. A população inicial se constitui por uma parte construída com indivíduos criados aleatoriamente e outra parte construída através da heurística de inserção I1 de Solomon (1987), com quatro formas diferentes de inserir o primeiro cliente de cada rota. Na fase de recombinação, foram utilizados quatro tipos de crossover: uniforme, 2 pontos, heurístico e PMX, e um operador de mutação baseado em uma busca heurística. A cada geração foram aplicados princípios de elitismo e pós-otimização utilizando a heurística λ -interchange de Osman (1993).

4.1. Algoritmo Genético

Os algoritmos genéticos vêm sendo aplicados em diversos problemas de otimização desde 1975 com a publicação do livro *Adaptation in Natural and Artificial Systems* de John Holland. Esses algoritmos se baseiam em princípios da teoria da genética populacional, onde uma população aumenta em número através da reprodução e pode ser diversificada pela combinação genética e/ou por mutação. Esta teoria foi construída através da combinação dos estudos de Charles Darwin (darwinismo) através de 20 anos de observações e experimentos que culminou na publicação do livro *On the origin of species by means of natural selection* em 1859 e com o trabalho de Gregor Mendel desenvolvido em 1865 sobre os princípios básicos de herança genética (Linden, 2006).

Os algoritmos genéticos, baseados na evolução biológica, são capazes de explorar fatores ambientais e convergir para soluções ótimas, ou quase ótimas em níveis globais. Diferente das outras metaheurísticas, o algoritmo genético trabalha com o conceito de "população de soluções", que são geradas e melhoradas a cada iteração. Portanto, cada solução é tratada como um indivíduo da população e a ideia central é que aquele com maior capacidade de adaptação ao meio tenha maior chance de se reproduzir e de sobreviver.

A cada iteração, os indivíduos são avaliados e um valor denominado *fitness* é atribuído. Esse valor reflete o grau de aptidão que o indivíduo apresenta e determina se suas características são valiosas a ponto de serem combinadas com outros indivíduos para gerar descendentes.

O processo de seleção e recombinação (operadores genéticos) procura produzir novos descendentes para construir a próxima geração de indivíduos que tomarão o lugar dos "pais" na população. Nesta fase, o princípio de seleção deve buscar pais que garantam a qualidade das próximas soluções ao mesmo tempo em que tenta explorar todo o espaço de solução. Após o processo de seleção, os operadores genéticos são responsáveis por realizar as trocas de código genético entre dois indivíduos (no caso de cruzamento) e mutação dentro de um mesmo indivíduo, com objetivo de construir uma nova população.

A cada geração, os melhores indivíduos são classificados como elite e recebem tratamento diferenciado dos demais. Esses indivíduos não "morrem", ao invés disso, são transferidos para a próxima população fazendo com que a qualidade da melhor solução a cada geração seja no mínimo igual. Além disso, os indivíduos elite podem receber um tratamento de pós-otimização em que um procedimento de melhoria é empregado. O elitismo e a pós-otimização não são etapas básicas dos algoritmos genéticos, no entanto podem ser empregados à estrutura principal servindo como procedimentos auxiliares de melhoria.

A Figura 4.1 apresenta um fluxograma dos principais passos do algoritmo genético, incluindo os procedimentos de elitismo e de pós-otimização.

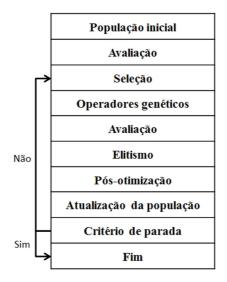


Figura 4.1 - Fluxograma dos principais passos do Algoritmo Genético (Linden, 2006)

4.2. Algoritmo Proposto

O algoritmo criado foi baseado na metaheurística algoritmo genético com cromossomos representados pela ordem de atendimento dos clientes sem delimitadores de rota. Uma parcela da população inicial foi gerada através de técnicas heurísticas conhecidas na literatura e a outra parcela gerada aleatoriamente. Na fase de recombinação, foram empregados quatro tipos de *crossover* e um tipo de mutação. Além disso, foram aplicadas técnicas de elitismo e de pósotimização.

A estratégia de solução do problema foi baseada em sete passos principais:

Passo AP-1Gerar população inicialPasso AP -2Avaliar todos os cromossomosPasso AP -3Selecionar os paisPasso AP -4Aplicar operadores genéticosPasso AP -5Aplicar elitismoPasso AP -6Aplicar pós-otimizaçãoPasso AP -7Testar condição de parada

O termo população se refere a um conjunto de indivíduos que por sua vez representa uma solução para o problema. O desempenho da busca e a qualidade da solução final encontrada pelo algoritmo genético estão ligados às características que a população inicial possui no início do processo. Os

indivíduos que compõe a população inicial devem conter esquemas diferenciados que contribuam para a diversidade da busca e esquemas com boas avaliações de aptidão. Para isso, na maioria das aplicações, a população inicial é formada por uma parte de indivíduos construídos através de heurísticas, geralmente métodos simples de roteamento conhecidos na literatura, e outra parte por indivíduos gerados de forma aleatória. Essa mistura de codificações enriquece o potencial da população inicial e aumenta as oportunidades de busca do algoritmo, prevenindo de ótimos locais ao mesmo tempo em que contribui para a convergência da solução. Além disso, o tamanho da população também é um dos parâmetros do algoritmo genético e esse número interfere diretamente no desempenho do processamento e na qualidade final da solução.

4.2.1 Geração da população inicial

Como mencionado anteriormente, a população inicial deve ser formada por uma mistura de cromossomos gerados de formas diferenciadas para assegurar que a busca resulte em soluções adequadas em relação ao objetivo do problema. Para isso, nessa pesquisa, uma parte dos cromossomos foi criada através da heurística de inserção I1 de Solomon (1987) denominada Heurística de Inserção *Push Forward (PFIH, Push Forward Insertion Heuristic*) e outra parte gerada aleatoriamente.

4.2.1.1 Push Forward Insertion Heuristic

A *PIFH* é um eficiente algoritmo de construção de rota proposto por Solomon (1987) que realiza inserções sequencias de clientes em uma rota. Solomon (1987) propôs três variações da heurística denominadas II, I2 e I3, que se diferenciam com relação ao cálculo do custo de inserção de um cliente na rota, composto por duas partes denominadas c_1 e c_2 , sendo que a variação I1 da *PFIH* apresentou melhores resultados no trabalho de Solomon (1987). Trabalhos como Potvin e Rosseau (1993) e Russel (1995) propõem uma adaptação da variação I1 da *PFIH* onde as rotas são construídas simultaneamente em relação ao critério de início da rota e às inserções. Segundo Potvin e Rosseau (1993), a heurística de inserção paralela é mais eficiente nos problemas onde os clientes são dispersos aleatoriamente e menos eficiente quando os clientes estão agrupados, em relação à variação I1 da *PFIH*. No entanto, para criar a solução inicial, foi utilizada a variação I1 da *PFIH*

por apresentar menor esforço computacional de processamento e menor complexidade de implementação.

O procedimento de construção de rota da *PFIH* começa com a inserção de um primeiro cliente que dá início a rota. Pelo fato da heurística realizar inserções em sequência, analisando apenas o cliente com menor custo de cada inserção, o critério de escolha do primeiro cliente que inicia a rota impacta diretamente a escolha da inserção dos próximos clientes. No trabalho de Solomon (1987) os critérios de escolha do primeiro cliente que obtiveram melhores resultados foram:

- O cliente mais distante do depósito, e
- O cliente com menor instante final da janela de tempo.

Por outro lado, Thangiah *et al.* (1994) utilizaram uma função ponderada que envolve as seguintes variáveis: tempo de viagem do cliente *i* até o depósito; instante final da janela de tempo do cliente *i*; e coordenada polar do cliente *i* normalizada em termos de tempo de viagem.

Nesse trabalho foram utilizadas as duas estratégias de inicialização de rota apresentadas por Solomon (1987) e uma equação ponderada baseada no trabalho de Thangiah *et al.* (1994), além da abordagem baseada na escolha aleatória. Para distinguir cada uma delas, foram adotadas as seguintes nomenclaturas:

- PFIH-DD: Push Forward Insertion Heuristic com cliente inicial sendo o mais distante do depósito.
- PFIH-JT: Push Forward Insertion Heuristic com cliente inicial sendo o que apresenta menor instante final da janela de tempo.
- PFIH-EQ: Push Forward Insertion Heuristic com cliente inicial sendo escolhido através da equação ponderada baseada na equação apresentada por Thangiah et al. (1994).
- PFIH-AL: Push Forward Insertion Heuristic com cliente inicial aleatório.

O custo de inicialização de cada cliente foi definido pela variável CustoInicialização $_i$ e a equação para cada caso são as seguintes:

-
$$PFIH-DD$$
: CustoInicialização_i = $-d_{0i}$ (20)

$$- PFIH-JT: CustoInicialização_i = l_i$$
 (21)

$$- PFIH-EQ: CustoInicialização_i = -\alpha d_{0i} + \beta l_i + \gamma [(|p_i - p_i|)d_{0i}]$$
 (22)

- PFIH-AL: $CustoInicialização_i = 1 + random(N-2)$ (23)

Onde:

- d_{0i} = distância de viagem do depósito até o cliente i;
- l_i = instante final da janela de tempo do cliente i;
- α = peso do tempo de viagem do cliente *i* ao depósito;
- β = peso para o instante final da janela de tempo do cliente i;
- γ = peso da coordenada polar normalizada pelo tempo de viagem;
- p_i = coordenada polar cliente i;
- p_i = coordenada polar do último cliente visitado na última rota formada, e
- N = número total de vértices no qual o subconjunto $\{1, 2, ..., n\}$ representa os clientes e o subconjunto $\{0, n+1\}$ representa o depósito.

Considerando que o objetivo é reduzir o custo de inicialização, a equação (20) dá prioridade ao cliente mais distante do depósito. A equação (21) prioriza os clientes com menor instante final da janela de tempo. A equação (22) foi desenvolvida com base no trabalho de Thangiah $et\ al.$ (1994), a equação original dá prioridade em primeiro lugar ao cliente mais distante do depósito, em seguida ao cliente com instante final da janela de tempo mais antecipada e por último ao cliente com menor valor de coordenada polar normalizada pela distância até o depósito, sendo que esse último critério se comporta como um processo de varredura em sentido horário. Nesse trabalho, essa parcela de custo, definida originalmente como $\gamma[(|p_i/360|)d_{0i}]$ foi substituído por $\gamma[(|p_j-p_i|)d_{0i}]$ que dá prioridade ao cliente mais próximo do último cliente inserido na última rota criada (sem valor para a primeira rota). Dessa forma, para clientes com valores iguais nos dois primeiros critérios, a equação fará com que o início de cada rota fique mais próximo do final da última rota criada, fazendo com que as rotas tenham mais adesão entre si nas posteriores operações de cruzamento e melhoria. Os valores dos pesos associados foram aproveitados do trabalho de Thangiah $et\ al.$ (1994) que os definiu empiricamente da seguinte forma: $\alpha=0.7$; $\beta=0.2$; $\gamma=0.1$. Finalmente a equação (23) escolhe um cliente de forma aleatória.

Após inserir o primeiro cliente na rota, através de um dos quatro métodos de inicialização apresentados, o resultado é uma rota parcial (i_0, i_1, i_{n+1}) onde i_1 é o cliente escolhido na inicialização (o que apresenta menor custo de inicialização) e i_0 e i_{n+1} representam o depósito.

O passo seguinte da heurística é composto por inserções sequenciais de clientes na rota sem violar as restrições de janela de tempo dos clientes e capacidade máxima do veículo. Para isso, um custo

de inserção é calculado para inserir cada cliente não roteirizado denominado u entre dois clientes sucessivos i e j.

O custo de inserção é composto por duas parcelas $c_1(i,u,j)$ e $c_2(i,u,j)$. O objetivo do primeiro critério é determinar a melhor posição na rota para inserir o cliente u. Para isso, para cada cliente não roteirizado, calcula-se a melhor posição p possível, definida por $c_1(i(u),u,j(u))$.

$$c_1(i(u), u, j(u)) = \min \left[c_1(i_{p-1}, u, i_p)\right], p = 1, ..., m$$
 (24)

Onde:

$$c_1(i, u, j) = \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j), \quad \alpha_1 + \alpha_2 = 1$$

$$\alpha_1, \alpha_2 \ge 0 \tag{25}$$

$$c_{11}(i, u, j) = d_{iu} + d_{uj} - ud_{ij}, \quad u \ge 0$$

$$c_{12}(i, u, j) = b_{iu} - b_{j}$$
(26)

Onde: b_{j_u} é o tempo atualizado que inicia o serviço do cliente j após a inserção do cliente u na rota.

A função c_{11} corresponde ao acréscimo de distância após a inserção do cliente u. A função c_{12} é a diferença entre o novo instante de início de serviço do cliente j e o instante anterior, em função da inserção do cliente u. A função c_1 , portanto, corresponde ao acréscimo de distância e tempo causado pela inserção do cliente u. A melhor posição de inserção de um cliente é aquela que minimiza esse acréscimo, Belfiore (2006).

O objetivo do segundo critério é determinar o cliente u a ser inserido na rota entre os clientes i e j que maximize a função $c_2(i(u), u, j(u))$.

$$c_2(i(u^*), u^*, j(u^*)) = \max[c_2(i(u), u, j(u))]$$

$$c_2(i, u, j) = \lambda d_{0u} - c_1(i, u, j), \qquad \lambda \ge 0$$
(27)

A função c_2 é a diferença (utilizando um critério de ponderação λ) entre a distância direta da base ao cliente u e o acréscimo em tempo e distância causados pela inserção da função c_1 . Por exemplo, para dois clientes com mesmo valor para função c_1 , o mais distante da base seria inserido.

A Figura 4.2 apresenta o pseudocódigo do algoritmo *PFIH* desenvolvido.

Passo PFIH-1 Comece com uma rota vazia do depósito Passo PFIH-2 SE todos os clientes foram roteirizados ENTÃO Vá para o *Passo PFIH-9* PARA todos os clientes não roteirizados c_i FAÇA Calcule o custo CustoInicialização_{ci} de todos os clientes de acordo com um dos critérios de inicialização: (20), (21), (22) ou (23) Passo PFIH-3 Ordene os clientes de forma crescente através de $CustoInicialização_{c_i}$ Passo PFIH-4 Selecione o primeiro cliente c_i * da lista ordenada que seja factível em termos de tempo e capacidade do veículo Passo PFIH-5 Insira c_i^* na rota corrente e atualize a capacidade da rota Passo PFIH-6 PARA todos os clientes não roteirizados u FAÇA **PARA** todas as bordas $\{i, j\}$ na rota corrente **FAÇA** Calcule o custo de inserir cada cliente u não roteirizado entre i e j de acordo com (27) **Passo PFIH-7:** Selecione o cliente u^* não roteirizado e a borda $\{i^*, j^*\}$ SE a inserção de u^* entre i^* e j^* é factível em termos de tempo e restrições de capacidade ENTÃO Insira o cliente u^* entre i^* e j^* Atualize a capacidade da rota corrente Vá para o Passo PFIH-6 **SENÃO** Vá para o *Passo PFIH-8* Passo PFIH-8: Comece uma nova rota do depósito Vá para o Passo PFIH-2 Passo PFIH-9: Todos os clientes foram roteirizados, fim.

Figura 4.2 - Algoritmo PFIH desenvolvido

4.2.1.2 Algoritmo Split

Nesse trabalho, o termo indivíduo será substituído por cromossomo em algumas ocasiões, essa substituição é feita quando o texto se refere à forma e manipulação da representação da solução, por outro lado, usar o termo indivíduo faz sentido quando a idéia em questão tem ligação com a população.

Os cromossomos foram construídos através da representação baseada em ordem sem delimitadores de rota. Isso quer dizer que um cromossomo será composto pelo código dos clientes, na ordem em

que serão atendidos pelos veículos, sem qualquer caractere que represente o início e termino de duas rotas consecutivas. Essa representação é frequentemente utilizada em problemas de otimização combinatória por apresentar estrutura de armazenamento de informação mais enxuta que as tradicionais representações binárias, diminuindo o esforço computacional necessário.

Os cromossomos sem delimitadores de rota podem ser divididos de diversas maneiras resultando em diferentes rotas. Prins (2004) propôs um procedimento de particionamento, denominado *split*, que busca os delimitadores ótimos para um cromossomo em termos do custo total para o problema de roteirização de veículos. Seja um dado cromossomo S = (1, 2, ..., n), o procedimento utiliza um grafo auxiliar H = (X, A, Z), onde X são os nós $\{0, 1, 2, ..., n\}$; A contém os clientes (i, j) se a viagem de i até j for factível em termos de capacidade e custo; Z contém os custos Z_{ij} da viagem de i até j calculados pela equação (28).

$$\forall (i,j) \in A: z_{ij} = c_{0,i+1} + \sum_{k=i+1}^{j-1} (s_k + c_{k,k+1}) + s_j + c_{j,0} \le L,$$

$$\forall (i,j) \in A: \sum_{k=i+1}^{j} d_k \le D$$
(18)

Onde:

- c_{ij} = custo de viagem do cliente i ao cliente j;
- s_i = tempo de serviço no cliente i;
- L = tempo máximo de operação de cada veículo;
- d_k = demanda do cliente k, e
- D =capacidade máxima do veículo.

A solução ótima para S corresponde ao mínimo caminho de 0 até n em H.

A Figura 4.3 apresenta o procedimento através de um exemplo exposto por Prins (2004), onde a Figura 4.3 (I) representa o cromossomo $S=(a,\ b,\ c,\ d,\ e)$ com demandas 5, 4, 4, 2 e 7 respectivamente, $Q=10, L=\infty$ e tempo de serviço (f) igual a 0 para todos os clientes. O grafo H utilizado pelo procedimento é mostrado em (II), onde o tempo de viagem de cada par de nós está associado ao trecho correspondente na figura. Como exemplo, uma possível rota $(0,\ a,\ b,\ 0)$ em H teria um peso de 55.

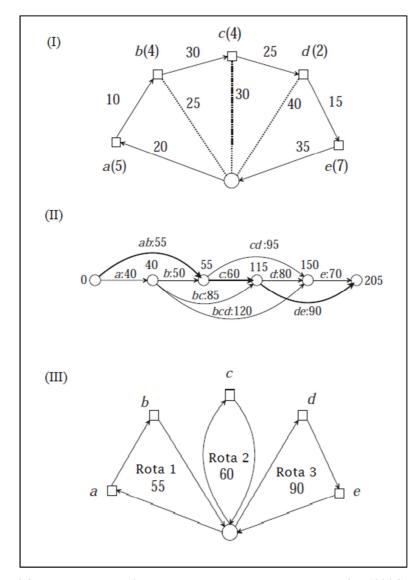


Figura 4.3 - Exemplo da criação de um cromossomo. Fonte: Prins (2004), p. 1989

A solução ótima para o exemplo é dado por três arcos (0, a, b, 0), (0, c, 0) e (0, d, e, 0) com custo total de 205, conforme o grafo (III) da Figura 4.3.

Prins (2004) fornece um algoritmo para o particionamento do cromossomo dividido em duas fases, rotulação e construção. Dado um cromossomo S = (1, 2, ..., n), dois rótulos são criados para cada nó j em S: V_j e P_j , onde V_j é o custo do mínimo caminho do nó 0 até o nó j em H e P_j é o predecessor de j nesse caminho. O algoritmo de rotulação possui um loop que a cada iteração enumera todas as possíveis subsequências factíveis em S e atualiza V_j e P_j . No final, o custo da mínima rota é dado em V_n . O vetor P é mantido para que as rotas sejam extraídas através do

algoritmo de construção. A Figura 4.4 apresenta o pseudocódigo do algoritimo de rotulação de Prins (2004) e a Figura 4.5 o algoritimo de construção do mesmo autor.

```
\begin{split} V_0 &= 0 \\ \mathbf{PARA} \text{ cada cliente } i \text{ em } S \text{ FAÇA} \\ V_i &= +\infty \\ \mathbf{PARA} \text{ cada cliente } i \text{ em } S \text{ FAÇA} \\ load &= 0; \cos t = 0; j = i \\ \mathbf{REPITA} \\ Load &= load + q_j \\ \mathbf{SE} \ (i == j) \ \mathbf{ENTÃO} \\ \cos t &= cost - c_{j-1,0} + c_{j-1,j} + s_j + c_{j,0} \\ \mathbf{SENÃO} \\ \cos t &= \cos t - c_{j-1,0} + c_{j-1,j} + s_j + c_{j,0} \\ \mathbf{SE} \ (load \leq Q) \ \mathbf{E} \ (\cos t \leq L) \ \mathbf{ENTÃO} \\ \mathbf{SE} \ (V_{i\cdot I} + \cos t < V_j) \ \mathbf{ENTÃO} \\ V_j &= V_{i\cdot I} + \cos t \\ P_j &= i - 1 \\ j &= j + 1 \\ \mathbf{ATÉ} \ (j \geq n) \ \mathbf{OU} \ (load \geq W) \ \mathbf{OU} \ (cost \geq L) \end{split}
```

Figura 4.4 - Pseudocódigo do algoritmo de rotulação, Prins (2004)

```
PARA cada cliente i em S FAÇA

trip(i) = Null
t = 0
j = n

REPITA

t = t + 1
i = P_j

PARA k = i + 1 ATÉ j FAÇA

Insira o cliente k no final da rota trip(t)
j = i

ATÉ i = 0
```

Figura 4.5 - Pseudocódigo do algoritmo de construção, Prins (2004)

No final do procedimento, cada *trip* representa uma rota com no mínimo um cliente e no máximo *n* clientes. O conjunto das *trips* representa a solução ótima para o particionamento do cromossomo *S*.

Entretanto, a formulação e os algoritmos desenvolvidos por Prins (2004) são destinados ao problema clássico de roteirização de veículos e, portanto, não consideram as janelas de tempo dos

clientes. Neste trabalho, foi utilizada uma modificação no método de Prins (2004) proposta por Cheng (2005) para tratar o problema de roteirização de veículos com janelas de tempo. Nesta abordagem, a formulação original apresentada em (28) é modificada da seguinte forma:

$$\forall (i,j) \in A: z_{ij} = c_{0,i+1} + \sum_{k=i+1}^{j-1} (w_i + s_k + c_{k,k+1}) + w_j + s_j + c_{j,0}$$
(19)

Onde: w_i é o tempo de espera para atender o cliente i.

Se $z_{ij} \leq L$, então o arco (i, j) existe no grafo auxiliar H e o mínimo caminho de 0 até n em H corresponde ao particionamento ótimo de S.

Para absorver essa modificação, Cheng (2005) apresenta o seguinte algoritmo de rotulação para o problema de roteirização de veículos com janelas de tempo, como mostra a Figura 4.6.

```
\begin{aligned} & V_0 = 0 \\ & \textbf{PARA} \text{ cada cliente } i \text{ em } S \text{ FAÇA} \\ & V_i = +\infty \end{aligned} \\ & \textbf{PARA} \text{ cada cliente } i \text{ em } S \text{ FAÇA} \\ & load = 0; cost = 0; j = i \\ & \textbf{REPITA} \\ & Load = load + q_j \\ & \textbf{SE} \ (i == j) \ \textbf{ENTÃO} \\ & cost = \max[c_{0,j}, e_j] + s_j + c_{j,0} \end{aligned} \\ & \textbf{SENÃO} \\ & cost = \max[cost - c_{j-1,0} + c_{j-1,j}, e_j] + s_j + c_{j,0} \end{aligned} \\ & \textbf{SE} \ (load \leq Q) \ \textbf{E} \ (cost \leq L) \ \textbf{ENTÃO} \\ & \textbf{SE} \ (V_{i,l} + cost < V_j) \ \textbf{ENTÃO} \\ & V_j = V_{i,l} + cost \\ & P_j = i - 1 \end{aligned} \\ & j = j + 1 \\ & \textbf{ATÉ} \ (j \geq n) \ \textbf{OU} \ (load \geq W) \ \textbf{OU} \ (cost \geq L) \end{aligned}
```

Figura 4.6 - Pseudocódigo do algoritmo de rotulação, Cheng (2005)

A principal alteração é o cálculo de custo que passa a considerar o instante inicial da janela de tempo no caso do veículo chegar adiantado no cliente. No entanto, a abordagem proposta por Cheng (2005) não assegura que os veículos cheguem antes do instante final da janela de tempo.

Para contornar isso, nesse trabalho foi incluída uma modificação no algoritmo descrito acima para evitar viagens com atraso, o algoritmo final utilizado é apresentado na Figura 4.7.

```
V_0 = 0
PARA cada cliente i em S FACA
     V_i = +\infty
PARA cada cliente i em S FAÇA
     load = 0; cost = 0; j = i
         REPITA
         Load = load + q_i
         SE (i == j) ENTÃO
              cost = max[c_{0, j}, e_j] + s_j + c_{j,0}
         SENÃO
            SE (cost - c_{j-1, 0} + c_{j-1, j} < l_i) ENTÃO
                  cost = max[cost - c_{j-1,0} + c_{j-1,j}, e_j] + s_j + c_{j,0}
             SENÃO
                  Infactivel = true
         SE (load \leq Q) E (cost \leq L) E (Infactivel = false) ENTÃO
             SE (V_{i-1} + cost < V_j) ENTÃO
            V_i = V_{i-1} + cost
            P_i = i - 1
            j = j + 1
         \mathbf{ATE}(j \ge n) \mathbf{OU}(load \ge W) \mathbf{OU}(cost \ge L)
```

Figura 4.7 - Pseudocódigo do algoritmo de rotulação utilizado nessa pesquisa

4.3. Seleção

A seleção é o processo que determina qual será o par de cromossomos que se reproduzirá e gerará descendentes. Esses cromossomos são denominados pais e são escolhidos mediante seu valor de *fitness*, através de um método pré-estabelecido.

Durante cada iteração, o princípio de seleção é aplicado a uma população de candidatos. Esse processo garante que os indivíduos com maior adaptação relativa tenham maiores chances de se reproduzir e passar seu código genético para as populações seguintes.

Nessa pesquisa foi utilizado o "método do torneio" como mecanismo de seleção por não favorecer diretamente os melhores indivíduos, incentivando a diversidade de características genéticas dentro da população. O princípio desse método é selecionar alguns indivíduos da população

aleatoriamente e fazer com que eles entrem em competição entre si, através da comparação do valor de *fitness* de cada um. O número de indivíduos selecionados para a competição é um parâmetro, denominado tamanho do torneio (k), que pode ser no mínimo igual a dois.

Segundo Linden (2006), não existe nenhum limite teórico para o valor máximo de k. Uma vez definidos os competidores, aquele que possui a melhor avaliação é selecionado para a aplicação do operador genético. Portanto, se k for igual ao tamanho da população (P), o método sempre selecionará o mesmo cromossomo (o melhor) e se forem escolhidos valores muito altos (próximos ao tamanho da população), os (n-P) indivíduos tenderão a predominar, uma vez que sempre um deles será o vencedor do torneio.

Um problema nesse método é que a única chance de um indivíduo com o menor valor de *fitness*, entre todos os indivíduos da população, ganhar o torneio é se a competição só o tiver como único candidato. No entanto, essa ocasião pode ocorrer, mas com uma probabilidade muito baixa, igual a $(n^P)^{-1}$, especialmente se n e P forem valores altos. Por isso, é mais comum que o valor de k seja em torno de dois, o que minimiza o problema. Nesse trabalho, no entanto, foi adotado o valor de k igual a quatro, como em Ombuki *et al.* (2006). Dessa forma, no caso do *crossover*, os dois melhores dentre esses quatro indivíduos são selecionados para a reprodução e no caso da mutação apenas um, o melhor.

4.4. Avaliação

O *fitness* de um cromossomo é um valor atribuído que reflete sua aptidão em relação ao meio. Esse valor é calculado através de uma função de avaliação que pode ser simplesmente um somatório de distâncias ou ainda um conjunto de parâmetros ponderados. A função de avaliação deve conter os objetivos envolvidos no problema. No caso do problema de roteirização de veículos, a função irá buscar minimizar o número de veículos utilizados e a distância total percorrida.

Alguns trabalhos incluem as restrições do problema na função de avaliação. Thangiah (1995), por exemplo, apresenta uma função de avaliação que procura minimizar o tempo de viagem penalizando violações de veículo sobrecarregado, atraso e tempo de rota excedido. Nessa função o modelo matemático é relaxado na capacidade do veículo, tempo de rota e janelas de tempo. Dessa forma, a busca aceita rotas que violem essas restrições, porém, é atribuído um alto custo, tornado-as menos interessante para a busca.

Outra forma de avaliar o cromossomo é utilizando o conceito multiobjetivo. Ghoseiri e Ghannadpour (2010) apresentaram uma proposta de avaliação de cromossomo baseado no método da fronteira de Pareto. Nessa abordagem é considerado que os dois objetivos principais do problema de roteirização (minimizar número de veículos e distância total) definem duas dimensões independentes em um espaço multiobjetivo de custo. Para tanto, são realizados testes de dominância em toda a população com relação aos objetivos considerados no problema e os cromossomos são divididos em camadas de estratificação. Os cromossomos da camada 1 são ditos não dominados e dominam os cromossomos da camada 2, os da camada 2 dominam os da camada 3 e assim sucessivamente. A ideia é que os melhores cromossomos se concentrem na camada 1 e que a solução final atenda de forma satisfatória todos os objetivos do problema.

Neste trabalho foi utilizada uma equação ponderada que envolve o número de veículos e a distância total percorrida com descrito em (28).

$$fitness_r = NV_r \times P_{N,VE \hat{I}CULOS} + DT_r \times P_{DIST \hat{A}NCIA}$$
 (28)

Onde:

- $fitness_i = \acute{e}$ o valor que representa a aptidão da rota r;
- NV_r = número de veículos utilizados na rota r;
- $P_{N.VEÍCULOS}$ = peso atribuído ao número de veículos utilizados;
- DT_r = distância total percorrida na rota r, e
- $P_{DIST\hat{A}NCIA}$ = peso atribuído à distância total percorrida.

4.5. Operadores Genéticos

Os operadores genéticos têm a função de diversificar a população mantendo características de adaptação adquiridas pelas gerações anteriores.

Nesse trabalho os cromossomos foram construídos utilizando a representação em ordem sem delimitadores de rota. Portanto, os conceitos apresentados nos próximos itens, sobre os operadores genéticos, serão feitos considerando essa representação.

4.5.1. Crossover

O operador *crossover* é o principal operador genético, pois sua função é criar vetores filhos através do cruzamento das características de dois vetores pais realizando trocas de segmentos de seus cromossomos. Esse cruzamento pode ainda se dar de diferentes maneiras, as formas mais conhecidas são os métodos baseados em pontos de corte adaptados dos procedimentos tradicionais desenvolvidos para as representações binárias.

Os métodos de pontos de corte, como apresentado por Thangiah (1995), e *PMX* proposto por Goldberg e Lingle (1985) são exemplos que se baseiam em pontos de corte e foram implementados nesse trabalho. Adicionalmente, os *crossovers* denominados uniforme e heurístico foram desenvolvidos nessa pesquisa, o primeiro foi baseado no trabalho de Linden (2006) que apresentou uma versão do operador para a representação binária e afirma que o *crossover* uniforme tende a obter resultados superiores em relação ao *crossover* dois pontos, o segundo realiza trocas de clientes de forma analítica onde o cromossomo filho resultante tem sempre o valor de *fitness* no mínimo igual ao cromossomo pai.

Na fase de recombinação genética, um operador de *crossover* é escolhido mediante sorteio através do método da roleta viciada. Nesse método, cada um dos operadores de *crossover* recebe uma percentagem diferente da roleta, fazendo com que aquele que obtiver maior valor tenha maior probabilidade de ser sorteado. Esse método influencia o sorteio de forma não eliminatória, uma vez que todos os cromossomos possuem chance de ser sorteado, ainda que sejam chances diferentes. A Figura 4.8 apresenta um exemplo do método da roleta viciada utilizado nesse trabalho.

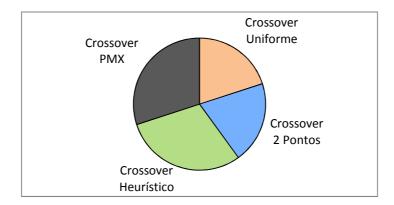


Figura 4.8 - Método da roleta viciada para escolha do crossover

44

A porção da roleta escolhida para cada *crossover* (P_{UNI} , P_{2PTS} , P_{HEU} e P_{PMX}) foi definida através de testes onde os valores foram variados e os resultados obtidos foram comparados entre si. Os resultados desses testes e análises estão descritas no capítulo 6 deste trabalho.

Os próximos itens descrevem o funcionamento dos operadores de cruzamento.

Crossover com dois pontos de corte

O cruzamento através do método de dois pontos de corte foi desenvolvido baseado em Linden (2006). O método escolhe duas posições em um vetor e então uma parte do código passa a ter a mesma ordem que a do outro vetor pai. Para exemplificar o procedimento, considere um conjunto de clientes {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, dois possíveis cromossomos P1 e P2 para esse conjunto seriam:

Os passos necessários para realizar o *crossover* com dois pontos de corte são:

Passo 1. Dois pontos de corte são definidos, nesse exemplo adotaremos corte_1 = posição 2 e corte_2 = posição 6.

Passo 2. O cromossomo filho (F) é criado com o código exatamente igual ao código do cromossomo P1 dentro dos dois pontos de corte.

Passo 3. Os clientes {6 3 1 8 10 5} que ficaram fora do intervalo definido pelos dois pontos de corte de P1 são recolocados em F na ordem que estão em P2, ou seja, {10, 8, 6, 5, 3, 1}.

Crossover PMX

O cruzamento *PMX* foi proposto por Goldberg e Lingle (1985) e também se baseia no princípio de pontos de corte. A abordagem consiste em definir dois pontos de corte nos cromossomos pais e permutar o código entre eles.

Esse procedimento pode ocorrer em cromossomos com genes repetidos. Neste caso, o valor repetido é substituído pelo gene pertencente à mesma posição do corte no outro cromossomo. A seguir é apresentado um exemplo do funcionamento do *crossover PMX* considerando os cromossomos pais: P1 (6 3 2 4 9 7 1 8 10 5) e P2 (10 9 8 7 6 5 4 3 2 1).

Passo 1. Dois cromossomos filhos são criados F1 e F2. O cromossomo P1 é copiado para F1 e o cromossomo P2 é copiado para F2.

F1 (6 3 2 4 9 7 1 8 10 5)

F2 (10 9 8 7 6 5 4 3 2 1)

Passo 2. Dois pontos de corte são definidos, nesse exemplo foi adotado corte_1 = posição 2 e corte_2 = posição 6.

F1 (6 3 | 2 4 9 7 | 1 8 10 5)

F2 (10 9 | 8 7 6 5 | 4 3 2 1)

Passo 3. Em seguida, trocam-se os genes (2 4 9 7) de F1 por (8 7 6 5) de F2.

F1 (6 3 | 8 7 6 5 | 1 8 10 5)

F2 (10 9 | 2 4 9 7 | 4 3 2 1)

Passo 4. Nota-se que os clientes 8, 6 e 5 no F1 e os clientes 2, 4 e 9 no F2 estão repetidos. Para corrigir esse problema, um processo iterativo é iniciado e a cada iteração, os clientes repetidos são substituídos pelos clientes na mesma posição dentro do corte, mas no outro filho.

46

Dessa forma, na primeira iteração, as trocas serão feitas da seguinte maneira: no cromossomo F1, cliente 8 pelo 2, cliente 6 pelo 9 e cliente 5 pelo 7; no cromossomo F2, cliente 2 pelo 8, cliente 4 pelo 7 e cliente 9 pelo 6.

Ainda será necessário repetir o processo descrito acima pois o cliente 7 está sendo repetido no F1 e no F2. As próximas trocas serão: no cromossomo F1, cliente 7 pelo 4; no cromossomo F2, cliente 7 pelo 5.

No final do procedimento o código entre os dois pontos de corte se manteve nos dois cromossomos filhos e apenas a parte exterior foi modificada.

Crossover Uniforme

O cruzamento uniforme apresentado foi baseado em Linden (2006) e pode ser descrito da seguinte forma: primeiramente o cromossomo filho F é criado como uma réplica de P1. Em seguida, para cada cliente no cromossomo F é sorteado um número entre 0 e 1. Se o número for menor que 0,5, o cliente é retirado do cromossomo e armazenado em um vetor auxiliar. Após percorrer todo o cromossomo, os clientes do vetor auxiliar são reinseridos no F na ordem em que se encontram no P2. Para exemplificar o procedimento, considere os cromossomos pais P1 (6 3 2 4 9 7 1 8 10 5) e P2 (10 9 8 7 6 5 4 3 2 1).

Passo 1. Criar F como réplica de P1.

Passo 2. Retirar cada cliente de F com uma probabilidade de 50% e inserir no vetor auxiliar V.

V (3 2 9 1 8 5)

Passo 3. Reinserir os clientes de V no F na ordem em que estão em P2, ou seja, {9, 8, 5, 3, 2, 1}.

F(69845732101)

Crossover Heurístico

O *crossover* heurístico se diferencia dos cruzamentos já descritos, pois se baseia em trocas analíticas de clientes entre os cromossomos pais. A ideia desse procedimento foi baseada no trabalho de Passarelli (2007) onde o filho gerado F recebe um cliente inicial de um dos pais e, a cada passo, é escolhido o próximo cliente a ser inserido que minimize a distância em relação ao último cliente do cromossomo.

Considerando os cromossomos pais P1 (6 3 2 4 9 7 1 8 10 5) e P2 (10 9 8 7 6 5 4 3 2 1), os passos do *crossover* heurístico são:

Passo 1. Criar os cromossomos filhos F1 e F2 onde F1 é uma réplica de P1 e F2 é uma réplica de P2.

F1 (6 3 2 4 9 7 1 8 10 5)

F2 (10 9 8 7 6 5 4 3 2 1)

Passo 2. Sortear uma posição aleatória nos cromossomos filhos, nesse caso posição 5.

F1 (6 3 2 4 9 | 7 1 8 10 5)

F2 (10 9 8 7 6 | 5 4 3 2 1)

Passo 3. Selecionar o primeiro cliente após o corte no cromossomo F1 (cliente 7). Em seguida, buscar esse cliente no F2 e trocar pelo primeiro cliente após o corte. Dessa forma os dois filhos terão o mesmo cliente após o corte.

F1 (6 3 2 4 9 | 7 1 8 10 5)

F2 (10 9 8 5 6 | 7 4 3 2 1)

48

Passo 4. As próximas trocas serão influenciadas pelo último cliente inserido em cada filho. No exemplo, os próximos clientes que serão analisados são os clientes 1 e 4 no F1 e F2 respectivamente. Dentre esses clientes, o cromossomo filho herdará aquele que minimiza a distância ao cliente escolhido no passo anterior. Supondo que $d_{74} < d_{71}$, escolhe-se o cliente 4 que permanece em sua posição no F2 e no F1 é substitui pelo cliente 1 utilizando o mesmo processo do Passo 3.

F1 (6 3 2 <u>1</u> 9 | 7 <u>4</u> 8 10 5)

F2 (10 9 8 5 6 | 7 4 3 2 1)

O procedimento descrito no Passo 4 é repetido até que todos os clientes tenham sido analisados. Nota-se que no final do processo temos apenas um cromossomo filho, pois F1 e F2 serão idênticos.

4.5.2. Mutação

O operador de mutação faz mudanças no material genético de um cromossomo. Esse operador é necessário para a introdução e manutenção da diversidade genética da população – assegura que a probabilidade de se chegar a qualquer ponto do espaço de busca nunca é zero, além de contornar o problema de mínimos locais, pois com este mecanismo altera-se levemente a direção da busca.

A escolha da probabilidade de mutação P_{MUT} é um parâmetro muito importante, já que valores altos conduzem a busca para uma convergência lenta, enquanto valores baixos podem conduzir a busca em direção a mínimos locais. Neste trabalho foi utilizado o operador de mutação heurístico, descrito no próximo item, e sua probabilidade P_{MUT} foi definida através de testes apresentados no capítulo 6 (Experimentos Computacionais) deste trabalho.

Mutação Heurística

A mutação heurística transfere um cliente para outra posição de forma analítica, se baseando na recolocação do cliente na melhor posição possível dentro do cromossomo. Considerando o cromossomo P1 (6 3 2 4 9 7 1 8 10 5), a mutação heurística segue os seguintes passos:

49

Passo 1. Criar o cromossomo filho F1 como réplica de P1.

F1 (6 3 2 4 9 7 1 8 10 5)

Passo 2. Selecionar um cliente em F1 aleatoriamente, nesse caso cliente 2.

F1 (6 3 <u>2</u> 4 9 7 1 8 10 5)

Passo 3. Calcular o custo de cada novo cromossomo criado na troca do cliente 2 por todos os outro clientes. Considerando que a troca do cliente 2 pelo cliente 7 é a que resulta em menor valor de *fitness*, os dois clientes são permutados.

F1 (6 3 7 4 9 2 1 8 10 5)

4.6. Elitismo

O elitismo é um recurso do algoritmo que, apesar de aumentar o tempo de processamento total, pode garantir que ao decorrer das gerações, a velocidade de convergência aumente.

A idéia do elitismo é que a cada iteração, os *k* indivíduos mais adaptados (com valor de *fitness* mais baixo) não "morram". Ao invés disso, eles devem passar para a próxima geração sem serem alterados. Isso garante que o desempenho do melhor indivíduo da próxima geração seja pelo menos igual ao da geração anterior fazendo com que o gráfico da avaliação do melhor indivíduo como função do número de gerações decorridas seja, no pior caso, uma função monotonamente crescente.

O número de cromossomos considerados elite é um parâmetro a ser considerado no algoritmo genético. Valores altos podem induzir a busca para ótimos locais. Por outro lado, valores baixos não trariam ganhos de desempenho que justificasse o uso desse método. Portanto, a taxa de cromossomos elite $P_{ELITISMO}$, foi definida como uma percentagem do número total de indivíduos da população.

4.7. Pós-Otimização

A etapa de pós-otimização não é necessariamente uma fase do algoritmo genético. Entretanto, trabalhos recentes como o de Ghoseiri e Ghannadpour (2010) vem mostrando a necessidade da combinação do algoritmo genético com técnicas de melhoria para buscar soluções mais adaptadas.

Nessa pesquisa, foi utilizada a estratégia λ -interchange como método de pós-otimização. Proposta por Osman (1993), a estratégia λ -interchange é um método de melhoria inter-rota que realiza trocas de clientes entre duas rotas distintas com o objetivo de buscar a melhoria das soluções já existentes.

4.7.1. λ -interchange

Dada uma solução para o problema (*VRPTW*) representada por $S = \{R_1, ..., R_p, ..., R_q, ..., R_k\}$ onde R_p é um grupo de clientes servidos por um veículo p, um λ -interchange entre os pares de rotas R_p e R_q é uma substituição entre os subconjuntos $S_1 \subseteq R_p$ de tamanho $|S_1| \le \lambda$ por outro subconjunto $S_2 \subseteq R_q$ de tamanho $|S_2| \le \lambda$ construindo duas novas rotas $R'_p = (R_p - S_1) \cup S_2$, $R'_q = (R_q - S_2) \cup S_1$ e uma nova solução $S' = \{R_1, ..., R'_p, ..., R'_q, ..., R_k\}$.

O número total de combinações para k rotas é k(k-1)/2. A busca seleciona todas as possíveis combinações de pares (R_p, R_q) como mostrado na sequência de pares (29).

$$(R_1, R_2), \dots, (R_1, R_k), (R_2, R_k), \dots, (R_{k-1}, R_k)$$
 (29)

O parâmetro λ indica o número máximo de clientes selecionados para a troca entre duas rotas. Para o caso em que o λ é igual à 2 (ou *2-Interchange*), os operadores de troca são (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1) e (2,2). Todos os operadores de troca são testados e o critério de parada é definido dentro de duas abordagens:

- First best (FB): a busca encerra quando uma a solução derivada de um dos operadores é melhor do que a solução atual, em relação ao cálculo do fitness, e
- Global best (GB): todas as combinações são testadas e a melhor de todas é escolhida.

Nesse trabalho, foi utilizado $\lambda = 2$ e a abordagem *global best*. Devido ao alto custo computacional do método, apenas os cromossomos elite foram melhorados com uma probabilidade $P_{POS-OTIMIZAÇÃO}$ de ocorrência.

4.8. Descrição dos parâmetros utilizados

Os parâmetros utilizados nesse trabalho foram divididos em três grupos principais: parâmetros genéticos, parâmetros da população inicial e parâmetros da função de avaliação.

Os parâmetros genéticos estão relacionados com as funcionalidades associadas à metaheurística algoritmo genético. Os parâmetros da população inicial definem como será constituída a população inicial. Os parâmetros da função de avaliação representam os pesos utilizados para ponderar o número de veículos e a distância total das soluções. Todos os parâmetros foram ajustados através de testes empíricos descritos na próxima seção com a finalidade de fazer com que a busca se adapte as características particulares das classes de problemas, melhorando o desempenho do algoritmo.

Em alguns casos a escolha desses parâmetros se constitui em decisões baseadas em *trade-off*, com relação à qualidade da solução e tempo de processamento. Para decidir qual deve ser o valor de cada parâmetro, muitos autores se baseiam em testes empíricos de seus algoritmos e nas experiências publicadas por outros autores.

A Tabela 4.1 apresenta os parâmetros que foram considerados nesta pesquisa.

Tabela 4.1 - Parâmetros considerados no algoritmo desenvolvido

Parâmetros Genéticos	População Inicial	Função de Avaliação
P _{TAMANHO DA POPULAÇÃO}	$P_{PFIH ext{-}JT}$	P _{N. VEÍCULOS}
P _{NÚMERO DE GERAÇÕES}	$P_{PFIH ext{-}DD}$	$P_{DIST\hat{A}NCIA}$
$P_{ELITISMO}$	$P_{PFIH ext{-}EQ}$	
$P_{POS ext{-}OTIMIZA ilde{CAO}}$	$P_{PFIH ext{-}\mathrm{AL}}$	
P_{UNI}	$P_{ALEAT\acute{O}RIO}$	
P_{2PTS}		
P_{HEU}		
P_{PMX}		
P_{MUT}		

A seguir é apresentada uma breve descrição de cada parâmetro utilizado no algoritmo desenvolvido.

Tamanho da população

Com uma população pequena o desempenho pode cair, pois desse modo a população fornece uma pequena cobertura do espaço de busca do problema. Uma grande população geralmente fornece uma cobertura representativa do problema. Porém são necessários maiores recursos computacionais, ou um maior tempo de compilação.

Critério de Parada

Foi utilizado um número máximo de gerações como critério de parada. Esse número interfere diretamente no tempo total de busca e deve ser ajustado de forma que garanta convergência do algoritmo sem interrupções prematuras da busca.

Elitismo

Com valor baixo em $P_{ELITISMO}$, a maior parte da população será substituída, podendo ocorrer perda de estrutura de alta aptidão. Com valor alto, a busca pode se tornar muito lenta.

Pós-otimização

O método λ -Interchange trata-se de uma heurística de melhoria, portanto se a probabilidade de ocorrência definida pelo parâmetro $P_{PÓS-OTIMIZAÇÃO}$ for muito alta, a busca pode cair em ótimos locais, além de aumentar o tempo de compilação, pela adição do custo computacional do método. Por outro lado, com valores baixos, a estratégia de pós-otimização pode não contribuir com a busca.

Crossover

A ocorrência de um determinado crossover está ligada à porção da roleta viciada que ele ocupa, definido pelos parâmetros P_{UNI} , P_{2PTS} , P_{HEU} e P_{PMX} , como descrito na seção 4.5.1. Em geral, crossovers tem grande capacidade de quebrar e gerar novas estruturas de cromossomos de formas diferentes. Nesse trabalho, quatro tipos de crossovers foram utilizados e o equilíbrio dos

parâmetros de ocorrência sobre a roleta viciada de cada um deles foi testado empiricamente, para escolher a configuração de parâmetros que trará melhores resultados.

Mutação

O parâmetro P_{MUT} define a probabilidade de ocorrência de mutação em relação ao *crossover*. Pelo fato da mutação ser realizada por um processo heurístico, descrito na seção 4.7.2., com uma taxa alta, a busca pode tender a ótimos locais. Por outro lado, com taxa baixa, é possível que uma boa troca não seja realizada.

População Inicial

A soma dos parâmetros $P_{PFIH-JT}$, $P_{PFIH-DD}$, $P_{PFIH-EQ}$, $P_{PFIH-AL}$ e $P_{ALEATÓRIO}$ definem a parcela da população inicial que será criada de forma heurística e aleatória, onde:

- $P_{PFIH-JT}$ = percentagem da população inicial criada pelo método PFIH-JT;
- $P_{PFIH-DD}$ = percentagem da população inicial criada pelo método PFIH-DD;
- $P_{PFIH-EO}$ = percentagem da população inicial criada pelo método PFIH-EQ;
- P_{PFIH-AL} = percentagem da população inicial criada pelo método *PFIH-AL*, e
- $P_{ALEATÓRIO}$ = percentagem da população inicial criada de forma aleatória.

O balanço entre soluções geradas de forma heurística e soluções geradas aleatoriamente na população inicial está diretamente ligado com a qualidade da solução final.

Função de Avaliação

No caso do problema de roteirização, a função de avaliação leva em consideração o número de veículos utilizados e a distância total percorrida. Os valores de cada um dos pesos podem ser ajustados para que a busca priorize uma determinada configuração de solução. Nesse trabalho, a prioridade se dá ao peso com valor mais próximo ao zero, uma vez que o algoritmo busca minimizar o valor final da função de avaliação.

Para utilizar a melhor combinação de valores dos parâmetros descritos a cima, testes computacionais foram realizados com o algoritmo desenvolvido recebendo diferentes valores prédefinidos. Tais testes foram realizados em cada grupo de parâmetros isoladamente enquanto os parâmetros dos outros grupos eram fixados. É importante salientar que tais parâmetros possuem

forte correlação entre si e que, mesmo separando-os em grupos, ganhos de sinergia podem ter sido perdidos, uma vez que todas as combinações possíveis não foram testadas. Por outro lado, testar todas as combinações possíveis torna-se inviável, dado o número de combinações e o número de instâncias testadas.

Nesse sentido, a abordagem de testes utilizada nesse trabalho procurou cobrir as combinações mais intuitivas em cada grupo de parâmetros. Os dados de tais testes foram coletados e analisados e estão apresentados no capítulo 5.

5. EXPERIMENTOS COMPUTACIONAIS

Neste capítulo serão apresentados os testes computacionais realizados com o algoritmo descrito no capítulo 4. O computador utilizado possui processador Intel(R) Core(TM) i5 CPU M480 @ 2.67GHz, com 4,00 GB de memória RAM, sistema operacional: Windows 7 Professional (64 Bits). O código foi escrito em linguagem Java e compilado na plataforma Eclipse IDE.

5.1. Instâncias de Solomon (1987)

O algoritmo desenvolvido foi testado em um conjunto de problemas desenvolvidos por Solomon (1987). Os resultados obtidos utilizando esse conjunto têm servido como base de comparação entre os vários algoritmos apresentados na literatura. O conjunto é constituído por 56 instâncias, cada uma com 100 clientes e um depósito central. Os problemas são separados em 3 grupos de acordo com a localização dos clientes. No primeiro grupo, denominado R, os clientes são distribuídos de forma aleatória. No segundo, denominado C, os clientes são agrupados em conjuntos ou *clusters*. No último, denominado RC, há uma mistura das duas situações anteriores.

Esses grupos recebem ainda uma segunda divisão de acordo com a janela de tempo. Os grupos C1, R1 e RC1 têm janelas de tempo mais curtas, com duração entre 25% e 50% do tempo de funcionamento do depósito, já os grupos C2, R2 e RC2 possuem janelas de tempo maiores, com variação de 75% a 100%.

5.2. Perfis de Desempenho

Essa pesquisa se fez uso do gráfico denominado *Perfil de Desempenho*, desenvolvido por Dolan e Moré (2002), para apresentar os resultados dos testes realizados neste trabalho. Originalmente, a finalidade do gráfico era facilitar a comparação de um conjunto de algoritmos *S*, elaborados para resolver um conjunto de problemas *P*, com base em uma medida escolhida para análise. No entanto, nessa pesquisa, o conjunto *S* representa um conjunto de configurações com diferentes valores de parâmetros, utilizados pelo algoritmo desenvolvido. Essa abordagem será empregada para avaliar quais os valores de parâmetros que, quando aplicados ao algoritmo desenvolvido para resolver o conjunto *P* de problemas, resultam em melhores resultados.

Considere $t_{p,s}$ como sendo a variável que define o desempenho resultante do problema $p \in P$ utilizando a configuração $s \in S$, no caso do problema de roteirização, $t_{p,s}$ representa a distância total percorrida pelos veículos. A razão de desempenho do algoritmo s com relação ao problema p, representada por $r_{p,s}$, é dada pela razão entre a distância total obtida por s e a menor distância obtida pelos algoritmos utilizados para resolver o problema p.

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}}$$
 (30)

A razão de desempenho mostra o comportamento de um algoritmo na resolução de um determinado problema. Note que a razão é sempre maior ou igual a 1. No caso em que $r_{p,s} = 1$, tem-se que o algoritmo s obteve o melhor resultado para o problema p.

O desempenho global do algoritmo s em relação ao conjunto P é dado por:

$$P(\tau) = \frac{1}{n_p} |\{ p \in P : r_{p,s} \le \tau \}|$$
 (31)

Onde: n_p é o número de problemas em P.

O perfil de desempenho $P(\tau)$ é uma função que associa a fração de problemas resolvidos pelo algoritmo p dentro de um fator $\tau \in \mathbb{R}$ do valor da menor distância encontrada pelo conjunto de algoritmos. Para exemplificar, tomemos um algoritmo s e P(1,01)=0.9, então, para 90% dos problemas, a solução obtida por s não superou em 1% a melhor solução obtida pelos algoritmos utilizados no teste.

Para traçar o gráfico do perfil de desempenho é necessário variar τ para cada problema e algoritmo. Após isso, devem-se agrupar as curvas obtidas e tomar como melhor algoritmo aquele que obtiver a curva mais alta.

5.3. Definição dos valores dos parâmetros utilizados no algoritmo

Nesta seção serão apresentados os testes computacionais utilizados para definição dos valores dos parâmetros envolvidos no algoritmo desenvolvido. Para descobrir qual a melhor configuração de

valores, os métodos de pós-otimização e elitismo foram desligados para que os testes não absorvessem melhorias resultantes dessas partes do algoritmo, mascarando o resultado final e a comparação das configurações de valores.

Para comparar os resultados obtidos a partir das diferentes configurações, foram utilizados gráficos de perfil de desempenho onde os valores utilizados para construí-los foram resultantes da média das soluções encontras em três processamentos para uma mesma configuração.

Todos os testes foram realizados de maneira independente para cada grupo de problemas de Solomon (1987), são eles: C1, C2, R1, R2, RC1 e RC2. Essa abordagem permite que os parâmetros adotados tornem o algoritmo mais aderente as diferentes características de cada grupo de problemas.

O primeiro grupo de testes definiu os valores que representam a participação de cada *crossover* na etapa de geração de novos indivíduos a partir de cruzamento. O segundo grupo de testes definiu a divisão entre mutação e cruzamento na etapa em que os operadores genéticos são aplicados. O terceiro grupo de testes determinou os valores dos pesos relacionados ao número de veículos utilizados e a distância total percorrida, envolvidos na função de avaliação. O último grupo de testes definiu a parcela de indivíduos da população inicial que foram criados pelo método *Push Forward Insertion Heuristc*. Assumindo a premissa que os parâmetros relacionados ao tamanho da população e número de gerações melhoram a solução na proporção em que seus valores aumentam, tendo impacto apenas no tempo computacional necessário, não foram realizados testes para definição de suas taxas, sendo que seus valores foram considerados os mesmos para todos os grupos de teses.

Para definição dos parâmetros utilizados no algoritmo proposto, descritos acima, foram necessárias 88 horas de testes.

5.3.1. Crossover

Neste item serão apresentados os testes realizados para a definição das taxas de *crossover* P_{UNI} , P_{2PT} , P_{HEU} e P_{PMX} utilizadas no algoritmo. Além de desligar os métodos de pós-otimização e elitismo, o método PIFH e mutação também foram desligados para que se pudesse observar apenas

o efeito de melhoria atribuído cada uma das diferentes configurações de valores dos *crossovers*. A Tabela 3 apresenta os parâmetros utilizados nos testes.

Tabela 5.1 - Parâmetros utilizados nos testes para definição das taxas dos crossovers

Parâmetros Genéticos		População II	nicial	Função de Avaliação		
P _{TAMANHO DA POPULAÇÃO}	50	$P_{PFIH ext{-}JT}$	0,0	P _{N. VEÍCULOS}	100	
P _{NÚMERO DE GERAÇÕES}	50	$P_{PFIH ext{-}DD}$	0,0	$P_{DIST\hat{A}NCIA}$	0,1	
$P_{ELITISMO}$	0,0	$P_{PFIH ext{-}EQ}$	0,0			
$P_{POS\text{-}OTIMIZA ilde{CAO}}$	0,0	$P_{PFIH ext{-}AL}$	0,0			
P_{MUT}	0,0	$P_{ALEATÓRIO}$	1,0			

Como apresentado no item 4.5.1. o algoritmo utilizado nessa pesquisa realiza um sorteio para descobrir qual operação de cruzamento será realizada a cada iteração. As taxas de cada *crossover*, que definem a área da roleta viciada, são de grande importância, pois cada operador possui mecanismos diferentes para realizar a recombinação genética.

Nesse sentido, foram criadas 15 configurações diferentes para as taxas de *crossover* como mostra a Tabela 5.2. Os testes 1, 2, 3 e 4 foram realizados utilizando cada *crossover* independentemente. Nos testes 5, 6, 7, 8, 9 e 10 foram testadas as combinações de 2 *crossovers* com mesma chance de ocorrência. Os testes 11, 12, 13 e 14 testaram combinações de 3 *crossovers* com mesma chance de ocorrência e no teste 15 foram utilizados todos os *crossovers*.

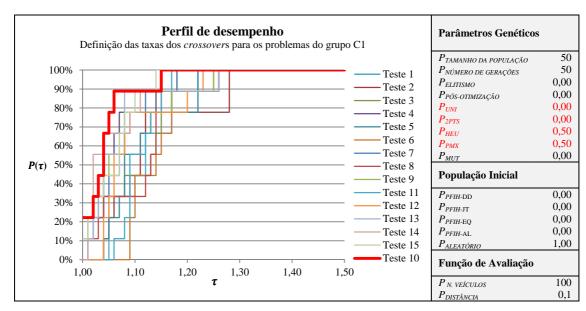
Tabela 5.2 - Configurações das taxas de crossover utilizadas nos testes

Teste 1		Teste 2		Teste 3		Teste 4		Teste 5	
P_{UNI}	1,00	P_{UNI}	0,00	P_{UNI}	0,00	P_{UNI}	0,00	P_{UNI}	0,50
P_{2PTS}	0,00	P_{2PTS}	1,00	P_{2PTS}	0,00	P_{2PTS}	0,00	P_{2PTS}	0,50
P_{HEU}	0,00	P_{HEU}	0,00	P_{HEU}	1,00	P_{HEU}	0,00	P_{HEU}	0,00
P_{PMX}	0,00	P_{PMX}	0,00	P_{PMX}	0,00	P_{PMX}	1,00	P_{PMX}	0,00

Teste 6		Teste 7		Teste 8		Teste 9		Teste 10)
P_{UNI}	0,50	P_{UNI}	0,50	P_{UNI}	0,00	P_{UNI}	0,00	P_{UNI}	0,00
P_{2PTS}	0,00	P_{2PTS}	0,00	P_{2PTS}	0,50	P_{2PTS}	0,50	P_{2PTS}	0,00
P_{HEU}	0,50	P_{HEU}	0,00	P_{HEU}	0,50	P_{HEU}	0,00	P_{HEU}	0,50
P_{PMX}	0,00	P_{PMX}	0,50	P_{PMX}	0,00	P_{PMX}	0,50	P_{PMX}	0,50

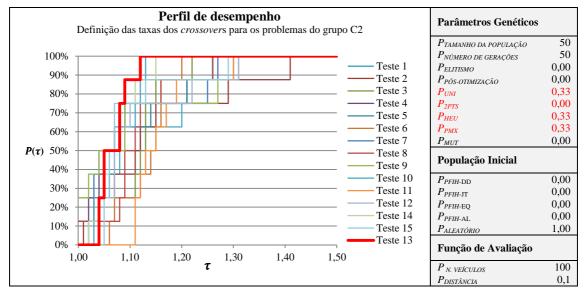
Teste 11		Teste 12		Teste 13		Teste 14		Teste 15	
P_{UNI}	0,33	P_{UNI}	0,33	P_{UNI}	0,33	P_{UNI}	0,00	P_{UNI}	0,25
P_{2PTS}	0,33	P_{2PTS}	0,33	P_{2PTS}	0,00	P_{2PTS}	0,33	P_{2PTS}	0,25
P_{HEU}	0,33	P_{HEU}	0,00	P_{HEU}	0,33	P_{HEU}	0,33	P_{HEU}	0,25
P_{PMX}	0,00	P_{PMX}	0,33	P_{PMX}	0,33	P_{PMX}	0,33	P_{PMX}	0,25

Os Quadros 5.1 a 5.6 apresentam os perfis de desempenho das configurações apresentadas na Tabela 4 aplicadas a cada grupo de problemas, seguido dos parâmetros que foram fixados. Nos gráficos, o eixo x representa a variação do fator τ e o eixo y representa o valor do perfil de desempenho $P(\tau)$.



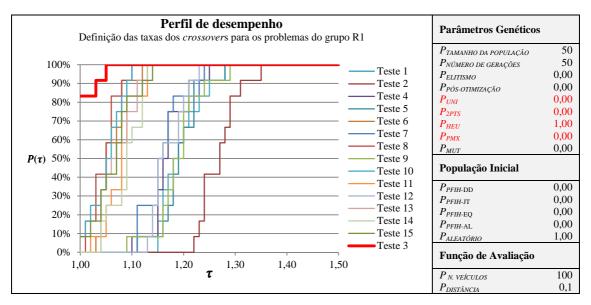
Quadro 5.1 - Curvas de perfil de desempenho para definição das taxas dos *crossovers* para os problemas do grupo C1

O gráfico de curvas de perfil de desempenho apresentado no Quadro 1 mostra que o Teste 10 teve melhor desempenho para o grupo C1 de problemas. Isso indica que os *crossovers* heurístico e *PMX* apresentam melhor aderência aos problemas testados e por isso devem ter probabilidades de ocorrência mais alta. Com isso, as taxas de *crossover* para os problemas do grupo C1 foram definidas com os seguintes valores: $P_{UNI} = 0,1$, $P_{2PT} = 0,1$, $P_{HEU} = 0,4$ e $P_{PMX} = 0,4$.

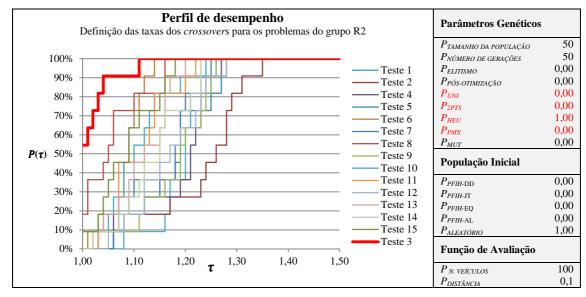


Quadro 5.2 - Curvas de perfil de desempenho para definição das taxas dos *crossovers* para os problemas do grupo C2

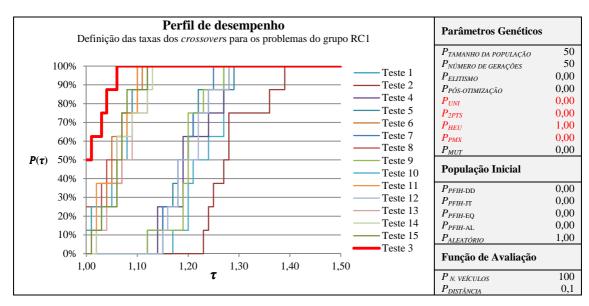
No grupo de problemas C2 a configuração do Teste 13 apresentou melhores resultados, indicando que os *crossover* uniforme, heurístico e *PMX* devem ser incentivados. Com isso, as taxas de *crossover* para os problemas do grupo C2 foram definidas com os seguintes valores: $P_{UNI} = 0.3$, $P_{2PT} = 0.1$, $P_{HEU} = 0.3$ e $P_{PMX} = 0.3$.



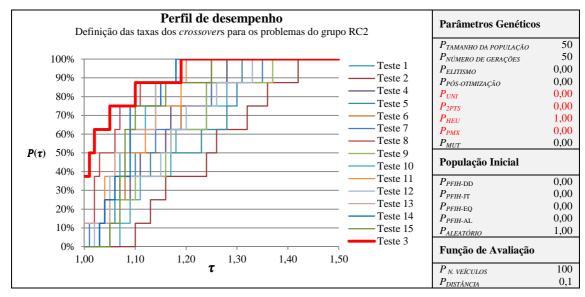
Quadro 5.3 - Curvas de perfil de desempenho para definição das taxas dos *crossovers* para os problemas do grupo R1



Quadro 5.4 - Curvas de perfil de desempenho para definição das taxas dos *crossovers* para os problemas do grupo R1



Quadro 5.5 - Curvas de perfil de desempenho para definição das taxas dos *crossovers* para os problemas do grupo RC1



Quadro 5.6 - Curvas de perfil de desempenho para definição das taxas dos *crossovers* para os problemas do grupo RC2

Os testes realizados nos grupos de problemas R1, R2, RC1 e RC2 apresentados nos Quadros 3, 4, 5 e 6, respectivamente, mostram uma forte predominância da configuração utilizada no Teste3, isso indica que para esses problemas o *crossover* heurístico deve ter maior probabilidade de ocorrência. Portanto, para os grupos de problemas R1, R2, RC1 e RC2 as taxas de *crossovers* foram definidas com os seguintes valores: $P_{UNI} = 0,1$, $P_{2PT} = 0,1$, $P_{HEU} = 0,7$ e $P_{PMX} = 0,1$.

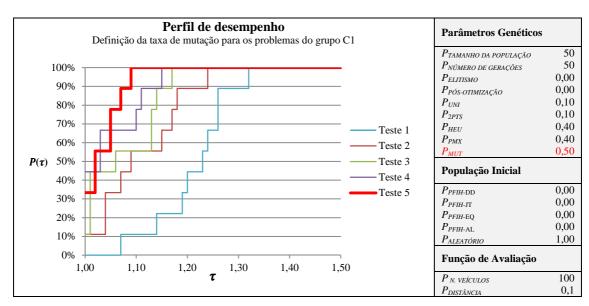
5.3.2. Mutação

Para definir qual a taxa P_{MUT} que traria melhores resultados na aplicação do algoritmo em cada grupo de problemas, foram realizados cinco testes utilizando as melhores configurações das taxas dos *crossovers* encontradas no item anterior, com a taxa P_{MUT} variando de 0,1 à 0,5. A Tabela 6 apresenta as configurações testadas.

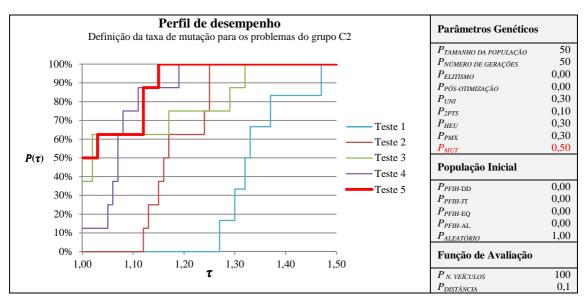
Tabela 5.3 - Configurações das taxas de mutação utilizadas nos testes

Teste 1		Teste 2		Teste 3		Teste 4		Teste 5	
P_{MUT}	0,10	P_{MUT}	0,20	P_{MUT}	0,30	P_{MUT}	0,40	P_{MUT}	0,50

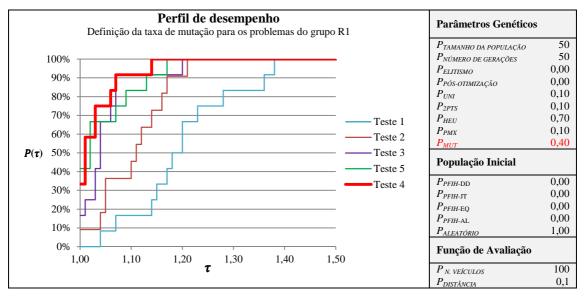
Os Quadros 5.7 à 5.12 apresentam os perfis de desempenho das configurações apresentadas na Tabela 6 aplicadas a cada grupo de problemas.



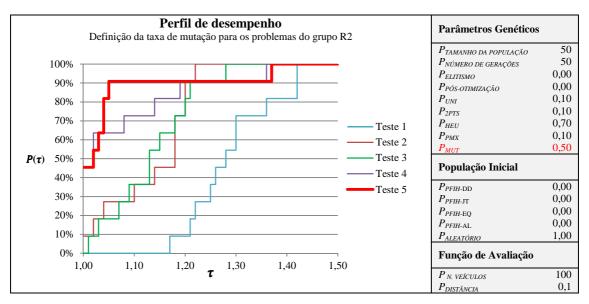
Quadro 5.7 - Curvas de perfil de desempenho para definição da taxa de mutação para os problemas do grupo C1



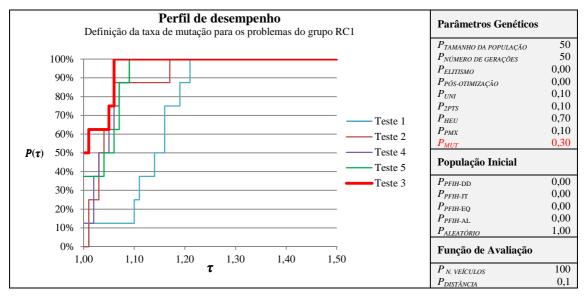
Quadro 5.8 - Curvas de perfil de desempenho para definição da taxa de mutação para os problemas do grupo C2



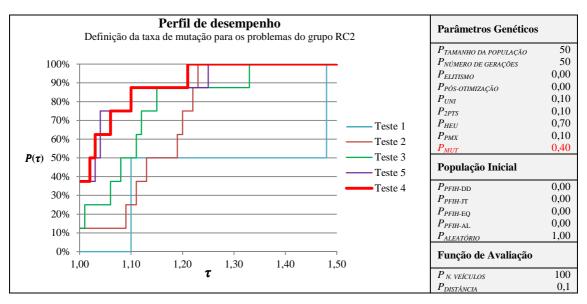
Quadro 5.9 - Curvas de perfil de desempenho para definição da taxa de mutação para os problemas do grupo R1



Quadro 5.10 - Curvas de perfil de desempenho para definição da taxa de mutação para os problemas do grupo R2



Quadro 5.11 - Curvas de perfil de desempenho para definição da taxa de mutação para os problemas do grupo RC1



Quadro 5.12 - Curvas de perfil de desempenho para definição da taxa de mutação para os problemas do grupo RC2

Em geral as taxas de mutação ficaram altas, sendo que os grupos C1, C2 e R2 ficaram com 0,5, RC2 e R1 ficaram com 0,4 e apenas o RC1 ficou com 0,3. Além disso, com exceção do grupo RC1, as curvas de desempenho com taxas 0,5 e 0,4 se apresentam muito próximas (como em C1, C2 e RC2, apresentadas nos Quadros 5.7, 5.8 e 5.12) indicando uma convergência da taxa de mutação nesse intervalo.

5.3.3. Push Forward Insertion Heuristic

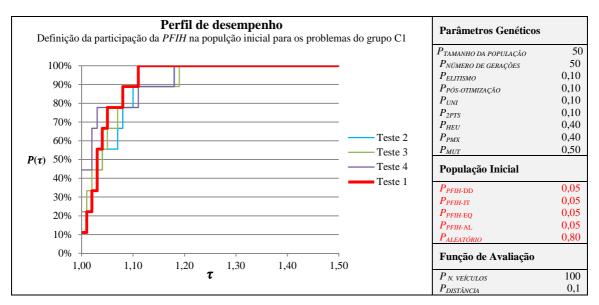
Nesse grupo de testes foi definido qual seria a parcela da população inicial criada a partir da heurística de construção I1 de Solomon (1987). As taxas de *crossover* e mutação foram consideradas conforme os resultados das análises realizadas nos itens anteriores.

Foram realizados quatro testes considerando uma variação da parcela da população inicial gerada pelo método *PFIH* de 20% a 80%. A Tabela 6 apresenta as taxas consideradas.

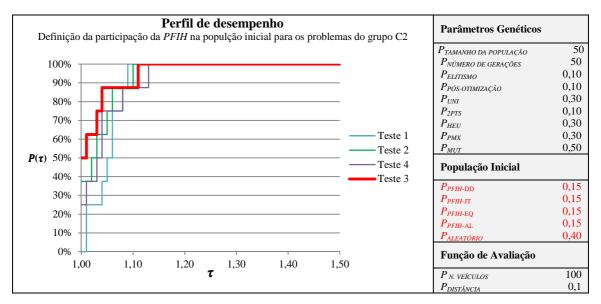
Tabela 5.4 - Configurações das taxas de PFIH utilizadas nos testes

Teste 1		Teste 2		Teste 3		Teste 4	
$P_{PFIH ext{-}DD}$	0,05	$P_{PFIH ext{-}DD}$	0,10	$P_{PFIH ext{-}DD}$	0,15	$P_{PFIH ext{-}DD}$	0,20
$P_{PFIH ext{-} m JT}$	0,05	$P_{PFIH ext{-} m JT}$	0,10	$P_{PFIH ext{-} m JT}$	0,15	$P_{PFIH ext{-} m JT}$	0,20
$P_{PFIH ext{-}EQ}$	0,05	$P_{PFIH ext{-}EQ}$	0,10	$P_{PFIH ext{-}EQ}$	0,15	$P_{PFIH ext{-}EQ}$	0,20
$P_{PFIH\text{-}AL}$	0,05	$P_{PFIH ext{-}AL}$	0,10	$P_{PFIH ext{-}AL}$	0,15	$P_{PFIH ext{-}AL}$	0,20

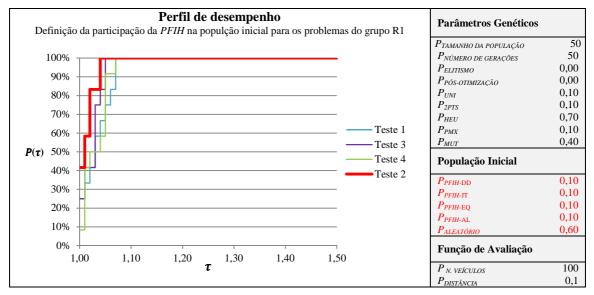
Os Quadros 5.13 à 5.18 apresentam os perfis de desempenho das configurações apresentadas na Tabela 5.4 aplicadas a cada grupo de problemas.



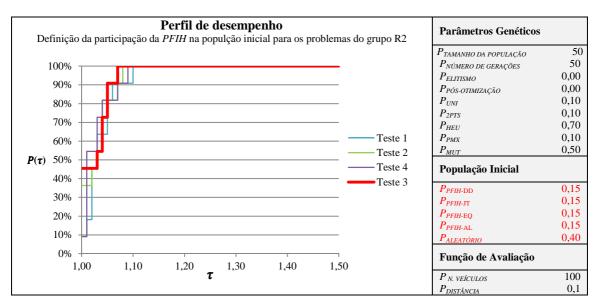
Quadro 5.13 - Curvas de perfil de desempenho para definição das taxas da heurística de construção *PFIH* para os problemas do grupo C1



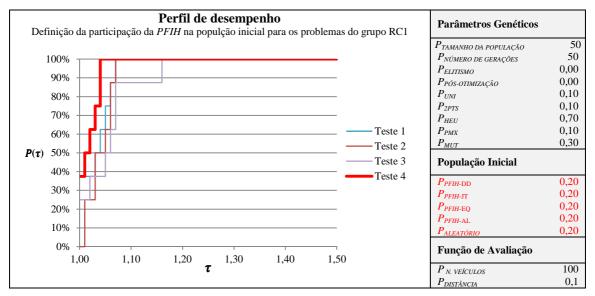
Quadro 5.14 - Curvas de perfil de desempenho para definição das taxas da heurística de construção *PFIH* para os problemas do grupo C2



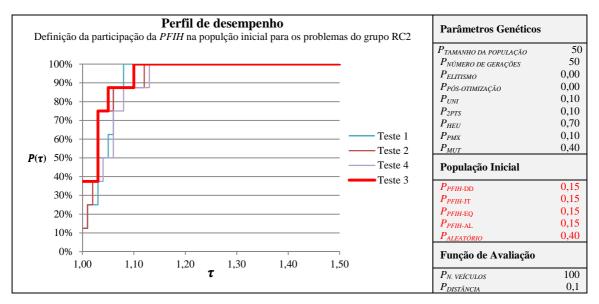
Quadro 5.15 - Curvas de perfil de desempenho para definição das taxas da heurística de construção *PFIH* para os problemas do grupo R1



Quadro 5.16 - Curvas de perfil de desempenho para definição das taxas da heurística de construção *PFIH* para os problemas do grupo R2



Quadro 5.17 - Curvas de perfil de desempenho para definição das taxas da heurística de construção *PFIH* para os problemas do grupo RC1



Quadro 5.18 - Curvas de perfil de desempenho para definição das taxas da heurística de construção *PFIH* para os problemas do grupo RC2

Os valores das configurações testadas que resultaram em menores distâncias para cada grupo de problemas foram adotados sem nenhuma variação. No grupo C1 de problemas, a taxa de *PFIH* foi igual a 20%, no grupo R1 igual a 40%, no grupo RC1 igual a 80% e nos grupos C2, R2 e RC2 igual a 60%.

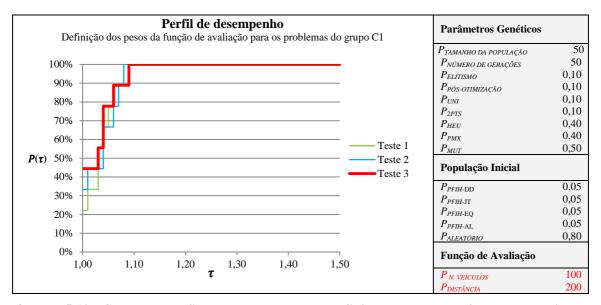
5.3.4. Função de Avaliação

O último grupo de testes teve o objetivo de aplicar três diferentes configurações de pesos na função de avaliação, a fim de observar qual seria a configuração mais adequada para cada um dos seis grupos de problemas: C1, C2, R1, R2, RC1 e RC2. A primeira configuração atribui um valor maior para o parâmetro referente ao número de veículos utilizados na solução ($P_{N. VEÍCULOS}$), em relação à distância total percorrida ($P_{DISTÂNCIA}$). A segunda configuração considera valores iguais para $P_{N. VEÍCULOS}$ e $P_{DISTÂNCIA}$. A terceira configuração é o inverso da primeira, onde o valor de $P_{DISTÂNCIA}$ é maior que o valor de $P_{N. VEÍCULOS}$. Os valores utilizados em cada teste são mostrados na Tabela 5.5.

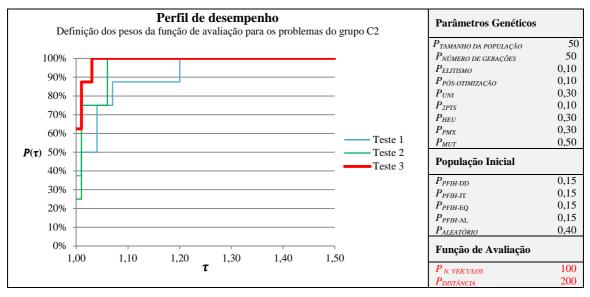
Tabela 5.5 - Configurações dos pesos da função de avaliação utilizada nos testes

Teste 1		Teste 2		Teste 3	
P _{N. VEÍCULOS}	100	P _{N. VEÍCULOS}	100	P _{N. VEÍCULOS}	100
$P_{DIST\hat{A}NCIA}$	0,1	$P_{DIST\hat{A}NCIA}$	100	$P_{DIST\hat{A}NCIA}$	200

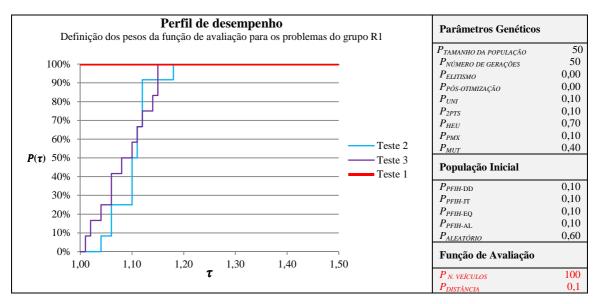
Os Quadros 5.19 a 5.24 apresentam os perfis de desempenho das configurações apresentadas na Tabela 5.5 aplicadas a cada grupo de problemas.



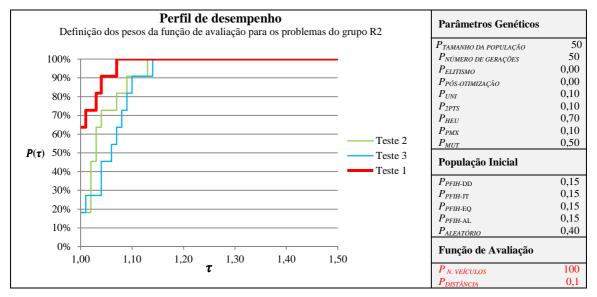
Quadro 5.19 - Curvas de perfil de desempenho para definição dos pesos da função de avaliação para os problemas do grupo C1



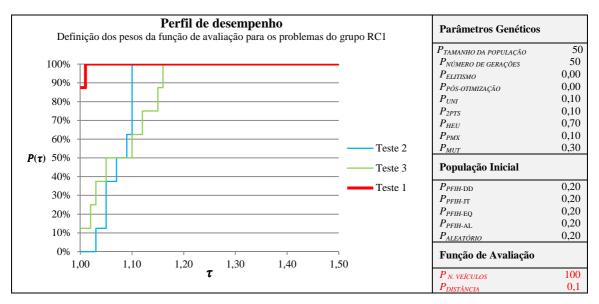
Quadro 5.20 - Curvas de perfil de desempenho para definição dos pesos da função de avaliação para os problemas do grupo C2



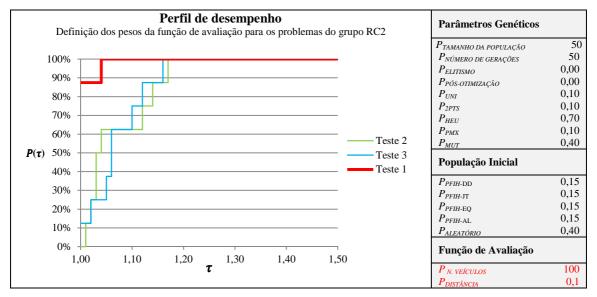
Quadro 5.21 - Curvas de perfil de desempenho para definição dos pesos da função de avaliação para os problemas do grupo R1



Quadro 5.22 - Curvas de perfil de desempenho para definição dos pesos da função de avaliação para os problemas do grupo R2



Quadro 5.23 - Curvas de perfil de desempenho para definição dos pesos da função de avaliação para os problemas do grupo RC1



Quadro 5.24 - Curvas de perfil de desempenho para definição dos pesos da função de avaliação para os problemas do grupo RC2

Os grupos de problemas C1 e C2 apresentaram menores distâncias quando utilizada a configuração do Teste 3, que atribui um peso maior para a distância total percorrida em relação ao número de veículos utilizados. Para os problemas dos grupos R1, R2, RC1 e RC2, a configuração com maior desempenho foi a do Teste 1 que prioriza diminuir o número de veículos em relação à distância total percorrida. Com isso, é possível observar que, para problemas que envolvem clientes agrupados em conjuntos, como na classe C de problemas, soluções com menores distâncias

percorridas são encontradas quando $P_{DISTÂNCIA}$ é maior que $P_{N.\ VEÍCULOS}$. Por outro lado, quando uma parte dos clientes é dispersa aleatoriamente, como na classe RC, ou ainda quando todos os clientes são dispersos aleatoriamente, como na classe R, soluções com menores distâncias são alcançadas quando $P_{DISTÂNCIA}$ é menor que $P_{N.\ VEÍCULOS}$.

5.4. Avaliação do algoritmo proposto

Nesta seção são apresentados os resultados obtidos com o algoritmo proposto nessa pesquisa quando aplicado aos problemas desenvolvidos por Solomon (1987), bem como a comparação com os melhores resultados publicados na literatura.

Os valores dos parâmetros encontrados nos testes descritos nos itens anteriores foram aplicados para cada grupo de problemas: C1, C2, R1, R2, RC1 e RC2, como apresentado na Tabela 5.6. Além disso, os parâmetros referentes ao tamanho da população e ao número de gerações foram fixados com valor igual a 100 de forma a viabilizar o processamento de todas as instâncias em relação ao tempo computacional.

Tabela 5.6 - Valores dos parâmetros utilizados no algoritmo proposto

	C1	C2	R1	R2	RC1	RC2
P _{TAMANHO DA POPULAÇÃO}	100	100	100	100	100	100
P _{NÚMERO DE GERAÇÕES}	100	100	100	100	100	100
$P_{ELITISMO}$	0,10	0,10	0,00	0,00	0,00	0,00
$P_{POS-OTIMIZA ilde{QAO}}$	0,10	0,10	0,00	0,00	0,00	0,00
P_{UNI}	0,10	0,30	0,10	0,10	0,10	0,10
P_{2PTS}	0,10	0,10	0,10	0,10	0,10	0,10
P_{HEU}	0,40	0,30	0,70	0,70	0,70	0,70
P_{PMX}	0,40	0,30	0,10	0,10	0,10	0,10
P_{MUT}	0,50	0,50	0,40	0,50	0,30	0,40
$P_{PFIH ext{-}DD}$	0,05	0,15	0,10	0,15	0,20	0,15
$P_{\it PFIH-JT}$	0,05	0,15	0,10	0,15	0,20	0,15
$P_{PFIH ext{-}EQ}$	0,05	0,15	0,10	0,15	0,20	0,15
$P_{PFIH ext{-}\mathrm{AL}}$	0,05	0,15	0,10	0,15	0,20	0,15
$P_{ALEAT \acute{O}RIO}$	0,80	0,40	0,60	0,40	0,20	0,40
P _{N. VEÍCULOS}	100	100	100	100	100	100
$P_{DIST\hat{A}NCIA}$	200	200	0,10	0,10	0,10	0,10

O indicador adotado para comparar os resultados obtidos nessa pesquisa com os melhores resultados publicados na literatura foi a distância total percorrida pelos veículos. Essa comparação foi realizada de duas formas diferentes: a primeira em relação aos resultados publicados obtidos por métodos exatos, utilizando apenas uma casa decimal e a segunda com relação aos resultados publicados obtidos por métodos heurísticos, utilizando duas casas decimais. Para os dois casos foram calculadas as discrepâncias (*gap*), entre o valor de referência e o valor obtido pelo algoritmo proposto, em relação ao número de veículos utilizados e em relação à distância total percorrida.

Tais informações são apresentadas para cada instância na Tabela 5.7 onde os resultados apresentados são resultantes da adoção da solução com menor distância total percorrida dentre três processamentos para cada instância de Solomon (1987) e os valores de referência para realizar a comparação são os mesmos apresentados na Tabela 2.2 da seção 2.4.

A Tabela 5.8 apresenta os valores médios das soluções obtidas e o respectivo *gap* em relação às soluções oriundas de métodos heurísticos (as soluções obtidas por métodos exatos foram suprimidas nessa tabela, pois algumas instâncias não possuem solução ótima até o momento).

Tabela 5.7 - Resultados obtidos utilizando o algoritmo proposto e comparação com os melhores resultados publicados na literatura

Instâncias		todos atos		Resultad	los Obtid	los*		Métodos eurísticos		Resultad	los Obtido	OS**	Tempo
Ilistancias	NV	DT	NV	<i>gap</i> [qtd.]	DT	<i>gap</i> [%]	NV	DT	NV	<i>gap</i> [qtd.]	DT	<i>gap</i> [%]	[min.]
C101	10	827,3	10	0	827,3	0,00%	10	828,94	10	0	828,94	0,00%	26,99
C102	10	827,3	12	2	981,6	18,65%	10	828,94	12	2	983,68	18,67%	30,69
C103	10	826,3	11	1	965,4	16,83%	10	828,06	11	1	967,22	16,81%	27,64
C104	10	822,9	12	2	973,5	18,30%	10	824,78	12	2	975,52	18,28%	29,52
C105	10	827,3	11	1	852,2	3,01%	10	828,94	11	1	853,89	3,01%	23,48
C106	10	827,3	12	2	886,5	7,16%	10	828,94	12	2	888,27	7,16%	33,47
C107	10	827,3	10	0	830,3	0,36%	10	828,94	10	0	835,32	0,77%	24,55
C108	10	827,3	11	1	882,9	6,72%	10	828,94	11	1	901,02	8,70%	31,03
C109	10	827,3	10	0	835,1	0,94%	10	828,94	10	0	836,71	0,94%	31,90
C201	3	589,1	3	0	589,1	0,00%	3	591,56	3	0	591,56	0,00%	6,87
C202	3	589,1	4	1	639,9	8,62%	3	591,56	4	1	642,58	8,62%	25,96
C203	3	588,7	4	1	670	13,81%	3	591,17	4	1	672,82	13,81%	31,53
C204	3	588,1	5	2	798,7	35,81%	3	590,6	5	2	801,61	35,73%	25,33
C205	3	586,4	3	0	586,4	0,00%	3	588,88	3	0	588,88	0,00%	20,89
C206	3	586,0	3	0	586,4	0,07%	3	588,49	3	0	588,88	0,07%	20,89
C207	3	585,8	3	0	586	0,03%	3	588,29	3	0	588,49	0,03%	36,83
C208	3	585,8	3	0	587,1	0,22%	3	588,32	3	0	589,61	0,22%	38,41

R101	20	1637,7	21	1	1728,1	5,52%	20	1642,87	21	1	1733,41	5,51%	33,46
R102	18	1466,6	18	0	1644	12,10%	18	1472,62	18	0	1648,85	11,97%	29,89
R103	14	1208,7	16	2	1421,8	17,63%	14	1213,62	16	2	1415,53	16,64%	30,31
R104	11	971,5	12	1	1159,6	19,36%	10	982,01	12	2	1152,53	17,36%	29,86
R105	15	1355,3	16	1	1457,1	7,51%	15	1360,783	16	1	1462,40	7,47%	27,43
R106	13	1234,6	15	2	1397,2	13,17%	13	1241,518	15	2	1402,02	12,93%	29,55
R107	11	1064,6	13	2	1280,6	20,29%	11	1076,125	13	2	1285,28	19,44%	27,66
R108	-	-	12	-	1165,7	-	10	948,573	12	2	1170,37	23,38%	22,46
R109	13	1146,9	14	1	1307,1	13,97%	13	1151,839	14	1	1311,70	13,88%	25,02
R110	12	1068,0	13	1	1275,3	19,41%	11	1080,36	13	2	1280,33	18,51%	26,94
R111	12	1048,7	13	1	1264,4	20,57%	12	1053,496	13	1	1269,28	20,48%	26,06
R112	-	-	11	11	1098,4	-	10	953,63	11	1	1103,38	15,70%	30,74
R201	8	1143,2	4	-4	1467,6	28,38%	9	1147,8	4	-5	1471,87	28,23%	43,12
R202	-	-	4	-	1308,5	-	5	1039,32	4	-1	1312,28	26,26%	37,86
R203	-	-	3	-	1169,1	-	5	874,87	3	-2	1173,09	34,09%	31,45
R204	-	-	4	-	882,8	-	3	735,8	4	1	887,09	20,56%	27,54
R205	-	-	3	-	1199,6	-	5	954,16	3	-2	1203,67	26,15%	36,65
R206	-	-	3	-	1110,9	-	4	884,25	3	-1	1114,94	26,09%	26,99
R207	-	-	3	-	1012,5	-	4	797,99	3	-1	1017,02	27,45%	32,22
R208	-	-	3	-	850,3	-	3	705,62	3	0	854,55	21,11%	36,63
R209	-	-	4	-	1102,8	-	5	860,11	4	-1	1107,15	28,72%	31,09
R210	-	-	4	-	1107,6	-	5	910,98	4	-1	1112,24	22,09%	35,40
R211	-	-	3	-	957,8	-	4	755,82	3	-1	962,37	27,33%	29,27
RC101	15	1619,8	17	2	1739,6	7,40%	15	1623,58	17	2	1743,54	7,39%	28,04
RC102	14	1457,4	16	2	1646,4	12,97%	14	1466,84	16	2	1650,40	12,51%	21,09
RC103	11	1258,0	13	2	1506,2	19,73%	11	1261,67	13	2	1510,23	19,70%	21,15
RC104	-	-	12	-	1295,6	-	10	1135,48	12	2	1299,34	14,43%	18,54
RC105	15	1513,7	16	1	1678,5	10,89%	16	1518,6	16	0	1682,98	10,82%	26,96
RC106	-	-	14	-	1505,7	-	13	1377,352	14	1	1509,45	9,59%	25,45
RC107	12	1207,8	14	2	1435,5	18,85%	12	1212,83	14	2	1439,20	18,66%	21,74
RC108	11	1114,2	13	2	1279	14,79%	11	1117,526	13	2	1283,27	14,83%	39,67
RC201	9	1261,8	4	-5	1589,4	25,96%	9	1266,11	4	-5	1593,51	25,86%	21,02
RC202	8	1092,3	4	-4	1418,8	29,89%	8	1096,75	4	-4	1422,27	29,68%	27,98
RC203	-	-	4	-	1205,5	-	5	926,89	4	-1	1209,93	30,54%	23,79
RC204	-	-	4	-	1036	-	4	786,38	4	0	1039,13	32,14%	19,40
RC205	7	1154,0	4	-3	1578,7	36,80%	7	1157,55	4	-3	1582,71	36,73%	24,34
RC206	-	-	4	-	1251,6	-	7	1056,21	4	-3	1255,03	18,82%	23,62
RC207	-	-	4	-	1171,6	-	7	966,08	4	-3	1175,22	21,65%	29,76
RC208	-	-	4	-	1037,9	-	4	779,84	4	0	1040,99	33,49%	29,93
* Desulted													_

^{*} Resultados considerando uma casa decimal ** Resultados considerando duas casas decimais

Tabela 5.8 - Média por classe dos resultados obtidos e dos melhores resultados publicados

	Método	s heurísticos		Resultados obtidos						
Classe	NV	DT	NV	<i>gap</i> [qtd.]	DT	<i>gap</i> [%]	Tempo [min.]			
C1	10,0	828,4	11,0	1,0	896,7	8,3%	28,81			
C2	3,0	589,9	3,5	0,5	633,1	7,3%	25,84			
R1	13,1	1.181,5	14,5	1,4	1.352,9	15,3%	28,28			
R2	4,7	878,8	3,5	-1,3	1.110,6	26,2%	33,47			
RC1	12,8	1.339,2	14,4	1,6	1.514,8	13,5%	25,33			
RC2	6,4	1.004,5	4,0	-2,4	1.289,8	28,6%	24,98			

O algoritmo proposto atingiu valores ótimos em instâncias das classes C1 e C2, em três oportunidades: C101, C201 e C205, além de valores próximos ao ótimo (considerando *gap* menor que 1%) nas instâncias C107, C109, C206, C207 e C208. Além disso, os valores de *gap* médio para essas duas classes (8,3% e 7,3% respectivamente) foram os mais baixos das seis classes de problemas testadas, confirmando assim o melhor desempenho do algoritmo em problemas com clientes distribuídos em conjuntos. Por outro lado, nas classes R1, R2, RC1 e RC2 nenhum problema foi resolvido de maneira ótima, mostrando que o algoritmo é menos eficiente em problemas onde os clientes apresentam características aleatórias com relação à localização.

Os problemas pertencentes às classes R2 e RC2 apresentaram os maiores valores de *gap* médio, em relação à distância total percorrida. Por esse motivo foram assumidos como os problemas em que o algoritmo proposto apresentou maior dificuldade de encontrar soluções que minimizasse a distância total percorrida. Por outro lado, se observou uma redução significativa no número de veículos utilizados, sendo que quando comparados aos resultados obtidos por métodos heurísticos essa redução se deu em 16 dos 27 problemas.

As principais características comuns às classes R2 e RC2 de problemas, além da distribuição aleatória dos clientes, é a duração da janela de tempo que é maior em relação aos problemas C1, R1 e RC1, com variação de 75% a 100% do tempo de funcionamento do depósito.

É provável que à medida que o número de iterações e/ou o tamanho da população são aumentados, melhores resultados devem ser atingidos, principalmente para os problemas das classes R2 e RC2, onde o baixo número de veículos utilizados é um indicativo que a busca não está estagnada e possíveis reduções nas distâncias de entrega podem ser atingidas através do aumento do número de veículos.

No entanto, cenários com aumento no número de iterações e tamanho da população foram inviabilizados pelo alto tempo computacional requerido para processar as 56 instâncias. De fato, para gerar os resultados apresentados na Tabela 5.7 foram necessárias 78,8 horas (3 rodadas de processamento de 26,27 horas cada).

6. CONSIDERAÇÕES FINAIS

Neste item serão apresentadas as principais conclusões extraídas das análises realizadas nessa pesquisa, bem como as recomendações e próximas etapas que darão continuidade ao desenvolvimento do componente de planejamento relacionado ao projeto do veículo autônomo.

6.1. Conclusões

Essa pesquisa apresenta uma abordagem para o problema de roteirização de veículos com janelas de tempo utilizando algoritmo genético. O estimulo para esse desenvolvimento foi apresentar um algoritmo eficaz de planejamento de rotas para uma frota de veículos autônomos, de forma que os mesmos realizem visitas em pontos específicos dentro de janelas de tempo estipuladas.

A revisão bibliográfica apresentou os principais métodos utilizados para solucionar o problema bem como importantes trabalhos publicados na literatura relacionados ao tema. Adicionalmente, a Tabela 2.2 é uma revisão atualizada dos melhores resultados, em relação ao objetivo de minimizar a distância total percorrida, publicados na literatura, destacando os métodos exatos e heurísticos e indicando os autores responsáveis.

A abordagem apresentada nessa pesquisa utilizou cromossomos representados pela ordem de atendimento dos clientes, com particionamento definido por um procedimento denominado *split*, baseado no trabalho de Prins (2004), desenvolvido originalmente para tratar o problema clássico de roteirização de veículos. Essa pesquisa apresentou uma modificação do algoritmo original, para tratar o *VRPTW*, que particiona os cromossomos em rotas de forma ótima considerando as janelas de tempo dos clientes. Essa abordagem se mostrou bastante eficiente, sendo que os particionamentos foram realizados de forma ótima sem adicionar tempo computacional considerável de execução.

Uma parte da população inicial foi construída através da heurística de inserção I1 de Solomon (1987), com quatro formas diferentes de inserir o primeiro cliente de cada rota, e outra parte gerada aleatoriamente. O objetivo foi diversificar as soluções criadas a partir da heurística de inserção I1 variando a forma de inicialização, da seguinte forma:

- O cliente mais distante do depósito (equação 20);

- O cliente com menor instante final da janela de tempo (equação 21);
- Equação ponderada proposta por Thangiah et al. (1994) (equação 22), e
- Escolha aleatória (equação 23).

De fato, as soluções obtidas por esses procedimentos adicionaram cromossomos com alto nível de aptidão, em termos de distância total, que somados às soluções geradas aleatoriamente, constituíram uma população inicial com qualidade suficiente para suportar as operações de busca do algoritmo.

O grande número de operadores genéticos disponíveis no algoritmo é um ponto relevante dessa pesquisa. Uma vez que a maioria das publicações apresenta abordagens utilizando em média dois operadores, nessa pesquisa foram utilizados quatro tipos de *crossover*: uniforme, 2 pontos, heurístico e *PMX*, e um operador de mutação baseado em busca local. A contribuição dessa diversidade dos operadores genéticos em relação à qualidade das soluções encontradas foi uma das apostas dessa pesquisa e de fato essa relação resultou em um algoritmo com maior capacidade de se adaptar as diferentes características dos problemas testados. No entanto, o estudo e desenvolvimento desses operadores, consumiram uma parte considerável do tempo da pesquisa.

Dois procedimentos que não fazem parte da estrutura dos algoritmos genéticos básicos foram incluídos nessa pesquisa. O primeiro foi o princípio de elitismo que assegura que os melhores cromossomos não sejam perdidos com o decorrer das gerações. O segundo foi uma etapa de pósotimização que utilizou uma heurística de melhoria denominada λ -Interchange de Osman (1993) aplicada aos cromossomos classificados como elite. A adição desses dois procedimentos foi vital para acelerar a convergência da busca, com reduções significativas na distância total percorrida.

Testes foram realizados para definir a melhor configuração dos diferentes parâmetros utilizados no algoritmo, são eles: parâmetros dos operadores genéticos (*crossover* e mutação), parâmetro que define a parcela da população inicial que deve ser criada a partir da heurística de construção *PFIH* e os pesos (para número de veículos utilizados e distância total percorrida) considerados na função de avaliação. Esses testes foram realizados de maneira independente para cara grupo de problemas (C1, C2, R1, R2, RC1 e RC2) de Solomon (1987) e a comparação entre as diferentes configurações de parâmetros foi realizada utilizando o gráfico de perfil de desempenho apresentado por Dolan e Moré (2002). Essa abordagem permitiu que os parâmetros adotados tornassem o algoritmo mais aderente as diferentes características de cada grupo de problemas.

Com relação aos resultados obtidos, nas classes de problemas C1 e C2, o algoritmo atingiu valores de *gap* médio, em relação à distância total, de 8,3% e 7,3% respectivamente, sendo os mais baixos das seis classes de problemas testadas, e valores ótimos em três instâncias: C101, C201 e C205, além de valores próximos ao ótimo (considerando *gap* menor que 1%) nas instâncias C107, C109, C206, C207 e C208.

As soluções encontradas para os problemas das classes R2 e RC2 apresentaram os maiores *gaps* de distância total percorrida em relação às melhores soluções publicadas na literatura. O motivo disso foi a maior duração das janelas de tempo desses problemas em relação aos problemas das classes C1, R1 e RC1, com variação de 75% a 100% do tempo de funcionamento do depósito. Por outro lado, em 16 dos 27 problemas pertencentes às classes R2 e RC2, foram encontradas soluções utilizando menor número de veículos. Isso indica que a busca não ficou estagnada e possíveis reduções nas distâncias de entrega, utilizando maior número de veículos, poderiam ser atingidas através do aumento do número de gerações.

No entanto, soluções com menores distâncias do que as que foram apresentadas na Tabela 5.7, principalmente para as classes R2 e RC2, não foram obtidas pelo reduzido número de gerações e/ou pelo tamanho da população, utilizados no algoritmo. Apesar do aumento desses valores teoricamente resultar em melhores soluções, o incremento no tempo computacional de processamento que isso representa inviabilizou tais testes. De fato, para gerar os resultados apresentados na Tabela 5.7 foram necessárias 78,8 horas (três rodadas de processamento de 26,27 horas cada).

Com isso, através da análise dos resultados obtidos nas classes C1 e C2 de problemas, é possível concluir que o algoritmo proposto possui maior facilidade para resolver problemas com clientes agrupados. Por outro lado, as soluções para os problemas pertencentes às classes R1, R2, RC1 e RC2 indicam que o algoritmo é menos eficiente em problemas em que os clientes apresentam algum grau de aleatoriedade em relação à distribuição geográfica. Por último, analisando as soluções das classes R2 e RC2, foi identificada uma dificuldade adicional em resolver problemas com clientes dispersos de forma aleatória e com grandes intervalos de janelas de tempo.

Por fim, dadas as vantagens e limitações que o algoritmo apresentou para diferentes características presentes em problemas de roteirização com janelas de tempo, descritas anteriormente, conclui-se que é possível utilizá-lo de forma eficiente para servir como o componente de planejamento de rotas em diversas aplicações envolvendo veículos autônomos.

6.2. Recomendações e próximas etapas

Apesar do algoritmo apresentado nessa pesquisa ser capaz de oferecer soluções para o problema de roteirização de veículos com janelas de tempo de maneira satisfatória, é possível ainda agregar melhorias para que soluções mais eficazes sejam atingidas.

Na fase de criação da população inicial foi utilizada a heurística de inserção II de Solomon (1987) com quatro formas diferentes de inserir o primeiro cliente na rota. Até a etapa final dessa pesquisa, se acreditou que essa abordagem forneceria soluções com qualidade suficiente para explorar diferentes locais do espaço de solução com boa qualidade. No entanto, após analisar os resultados apresentados no item 5 se observou que essa estratégia não funcionou de maneira adequada para os problemas com clientes dispersos de forma aleatória das classes R e RC. Uma tentativa para minimizar isso seria adicionar na população inicial cromossomos criados a partir da heurística de Potvin e Rosseau (1993) que, segundo os autores, apresenta melhores resultados para problemas com clientes dispersos aleatoriamente.

O tempo computacional também foi um limitante nessa pesquisa, uma vez que a principal motivação do estudo e teste das abordagens de solução esteve focada na qualidade do resultado final dos problemas, o que resultou em um algoritmo com grande potencial, provido com diferentes *crossovers* e métodos de melhoria, mas com tempo de execução elevado. No entanto, algoritmos genéticos em geral apresentam maior tempo computacional para resolver grandes instâncias dos problemas pertencentes à classe NP-*Hard* (como o caso das instâncias de Solomon (1987)), quando comparados a outras metaheurísticas como busca tabu e *simulated annealing* por exemplo. Uma revisão do algoritmo com enfoque no tempo computacional, sem perda de qualidade, será uma boa melhoria para o sistema. Isso possibilitará que mais gerações sejam feitas, podendo ainda melhorar os resultados obtidos.

As próximas etapas de pesquisa serão voltadas para a implantação do algoritmo apresentado em um sistema de controle de veículos autônomos onde o enfoque principal será gerar rotas para uma frota de veículos respeitando as janelas de tempo de visita, servindo como sistema de planejamento do sistema como um todo.

O algoritmo deverá ser executado em um computador de grande capacidade de processamento e deverá ser capaz de transferir as rotas geradas para os veículos através do componente de comunicação do sistema. Adicionalmente, integrações com o componente de navegação poderão ser realizadas para que alterações nas rotas sejam feitas em tempo real durante a execução das

tarefas. Para tanto, modificações deverão ser realizadas para incorporar tal dinamismo no algoritmo.

7. REFERÊNCIAS BIBLIOGRÁFICAS

ACHUTHAN, N.R.; CACCETA, L.; HILL, S.P. An improved branch-and-cut algorithm for the capacitated vehicle routing problem. **Transportation science**, v.37, n.2, p.153-169, 2003.

ALVARENGA, G.B.; MATEUS, G.R.; DE TOMI, G. A genetic and set partitioning two phase approach for the vehicle routing problem with time windows. **Computers & Operation Research**, v.34, p.1561–1584, 2007.

ANTES, J.; DERIGS, U. A New Parallel Tour Construction Algorithm for the Vehicle Routing Problem with Time Windows. Department of Economics and Computer Science, University of Köln, Germany, 1995, (Working Paper)

ARAÚJO, C.E.G. **Algoritmos genéticos híbridos sem delimitadores de rotas para problemas de roteirização de veículos**. 2008. 89 p. Tese (Mestrado) — Escola politécnica, Universidade de São Paulo, São Paulo, 2008.

ASSAD, A.A. Modeling and implementation issues in vehicle routing. In: VEHICLE ROUTING: METHODS AND STUDIES, 1988, North Holland, Amsterdam, p.7-46, 1988.

ATKINSON, J. B. A greedy look-ahead heuristic for combinatorial optimisation: An application to vehicle scheduling with time windows. **J. Operational. Research Society**, v.45, n.6, p.673-684, 1994.

BADEAU, P. ET AL. A parallel tabu search heuristic for the vehicle routing problem with time windows. **Transportation Research**, v.5, p.109-122, 1997.

BAKER, E.K. Vehicle routing with time windows constraints. **Logistic and Transportation Review**, v.18, n.4, p.385-401, 1982.

BAKER, J.E. Reducing bias and inefficiency in the selection algorithm. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS AND THEIR APPLICATION, 2, 1987, New Jersey. Anais... New Jersey: Lawrence Erlbaum Associates, 1987, v.1, p.14-21.

BAKER, B.M.; AYECHEW, M.A. A genetic algorithm for the vehicle routing problem. **Computers & Operations Research**, v.30, n.5, p.787-800, 2003.

BARBAROSOGLU, G.; OZGUR, D. A tabu search algorithm for the vehicle routing problem. **Computers & Operations Research**, v.26, n.3, p.255-270, 1999.

BARD, J.F.; KONTORAVDIS,G.; YU,G. A Branch and Cut Procedure for the Vehicle Routing Problems with Time Windows. **Transportation Science**, v.36, n.2, p.250-269, 2002.

BELFIORE, P.P. Scatter search para problemas de roteirização de veículos com frota heterogênea, janelas de tempo e entregas fracionadas. 2006. 203 p. Tese (Doutorado) — Escola politécnica, Universidade de São Paulo, São Paulo, 2006.

BENENSON, R., ET AL. Toward urban driverless vehicles. International Journal of Vehicle Autonomous Systems. Advances in autonomous Vehicle Technologies for Urban Environment, v.1, n.6, p.4–23, 2008.

BENENSON R., ET AL. Integrating Perception and Planning for Autonomous Navigation of Urban Vehicle. **IEEE/RSJ International Conference on Intelligent Robots and Systems** (**IROS**), Beijing, Chine 2006.

BENT, R.; HENTENRYCK, V.P. A two stage hybrid local search for the vehicle routing problem with time windows. **Transportation Science**, v.38, p.515-530.

BERGER, J.; BARKAOUI, M. A hybrid genetic algorithm for the capacitated vehicle routing problem In: E. Cantú-Paz, 2003, Chicago. Anais... Chicago: Springer-Verlag, 2003.

BODIN, L.D.; GOLDEN, B. Classification in Vehicle Routing and Scheduling. **Networks**, v.11, n.2, p.97-108, 1981.

BODIN, L.D., ET AL. Routing and scheuduling of vehicle and crews: The state of the art. **Computers & Operations Research**, v.10, n.2, p.63-211, 1983.

BRÄYSY, O.; GENDREAU, M. Vehicle Routing Problem with Time Windows, Part II: Metaheuristics, **Transportation Science**, v.39, n.1, p.119-139, 2005.

CHABRIER, A. Vehicle Routing Problem with Elementary Shortest Path based Column Generation. **Computers and Operations Research**, v.33 n.10, 2005.

CHENG, L. A Genetic Algorithm for the Vehicle Routing Problem with Time Windows. 2005. Tese (Mestrado) – Departament of Mathemarics and Statistics, University of North Carolina Wilmington, Wilmington, NC, 2005.

CHRISTOFIDES, N.; MINGOZZI A.; TOTH, P. The vehicle routing problem. In: Christofides N, Mingozzi A, Toth P, Sandi C, editors. **Combinatorial optimization**. New York: Wiley, p. 315-38, 1979.

CHRISTOFIDES, N.; MINGOZZI A.; TOTH, P. Exact Algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. **Mathematical programming**, v.20, p.255-282, 1981.

CHRISTOFIDES, N. Vehicle Routing. In: The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, Lawer, E.L.; Lenstra, J.K.; Kan, A.H.G.R.; Shmoys, D.B (Eds), John Wiley & Sons, 1985.

CLARKE, G.; WRIGHT, J.W. Scheduling of Vehicles from a Central Depot to a number of Delivery Points. **Operations Research**, v.12, p.568-581, 1964.

COOK, W. E RICH, J. L. A parallel cutting plane algorithm for the vehicle routing problem with time windows. Computational and Applied Mathematics, Rice University, Houston, TX, 1999, (Working Paper)

- CORDEAU, J. F.; LARPORTE, G.; MERCIER, A. A unified tabu search heuristic for vehicle routing problems with time windows. **Journal of the Operational Research Society** v.52, p.928-936, 2000.
- CUNHA, C.B. Uma contribuição para o problema de roteirização de veículos com restrições operacionais. 1997. 222 p. Tese (Doutorado) Escola Politécnica, Universidade de São Paulo, São Paulo, 1997.
- CUNHA, C.B., GUALDA, N.D.F. Heurísticas baseadas em relaxação lagrangiana para o problema de roteirização de veículos com restrições operacionais. In: CONGRESSO DE PESQUISA E ENSINO EM TRANSPORTES, 11, 1997, Rio de Janeiro. **Anais**... Rio de Janeiro: ANPET, 1997, v.2, p.843-855.
- CUNHA, C.B. Aspectos práticos da aplicação de modelos de roteirização de veículos a problemas reais. **Transportes**, v.8, n.2, p.51-74, 2000.
- CUNHA, C.B. Contribuição à Modelagem de Problemas em Logística e Transportes. 2006. 315 p. Tese (Livre Docência) Escola Politécnica, Universidade de São Paulo, São Paulo, 2006.
- CZECH, Z.J.; CZARNAS, P. Parallel simulated annealing for the vehicle routing problem with time windows. In: EUROMICRO WORKSHOP ON PARALLEL DISTRIBUTED AND NETWORK-BASED PROCESSING, 10, 2002, Canary Islands. **Anais...** p. 376-383, 2002.
- DANNA, E.; LE PAPE, C. Accelerating branch-and-price with local search: A case study on the vehicle routing problem with time windows. In: **Column Generation**, G. Desaulniers, J. Desrosiers, and M. M. Solomon (eds.), Kluwer Academic Publishers, 2005. p.99-130.
- DANTZIG, G. B. E RAMSER, R. H. The Truck Dispatching Problem. **Management Science**, v.6, p.80-91, 1959.
- DESROCHERS, M.J. ET AL. A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows. **Operations Research**. v.40, p.342-354, 1992.
- DOLAN, E.D.; MORÉ, J.J. Benchmarking optimization software with performance profiles. **Mathematical Programming**. v.A91, p.201-213, 2002.
- EKSIOGLU, B.; VURAL, A.V.; REISMAN, A. The vehicle routing problem: a taxonomic review. **Computers & Industrial Engineering**, v.57, p.1472-1483, 2009.
- FEO, T.A.; RESENDE, M.G.C. A probabilistic heuristic for a computationally difficult set covering problem. **Operations Research Letters**, v.8, p.67–71, 1989.
- FEO, T.A.; RESENDE, M.G.C. Greedy randomized adaptive search procedures. **Journal of Global Optimization**, v.6, p.109-133, 1995.
- FISHER, M.; JAIKUMAR, R. A generalized assignment heuristic for vehicle routing. **Networks**, v.11, p.109-124, 1981.

FISHER, M.L. ET AL. Vehicle routing with time windows: Two optimization algorithms. **Operations Research**, v.5, n.3, p.488-492, 1997.

FRIZZELL, P.W.; GIFFIN, J.W. The Split Delivery Vehicle Scheduling Problem with Time Windows and Grid Network Distances. **Computers & Operations Research**, v.22, n.6, p.655-667, 1995.

GAMBARDELLA, L. M.; TAILLARD, E.; AGAZZi. A multiple ant colony system for vehicle routing problems with time windows. In: **New Ideas in Optimization**, David Corne, Marco Dorigo, Fred Glover (Eds.), McGraw-Hill, London, 1999. p.63-76.

GARCIA, B.L.; POTVIN, J.Y.; ROUSSEAU, J.M. A Parallel Implementation of the Tabu Search Heuristic for Vehicle Routing Problems with Time Windows Constraints. **Computers & Operational Research**, v.21, p.1025-1033, 1994.

GENDREAU, M.; HERTZ, A.; LAPORTE, G. New Insertion and Postoptimization Procedures for the Traveling Salesman Problem. **Operation Research**, v.40, n.6, p.1086-1094, 1992.

GENDREAU, M; HERTZ, A.; LAPORTE, G. A tabu search heuristic for the vehicle routing problem. **Management Science**, v.40, n.10, p.1276-1290, 1994.

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

GILLETT, B.; MILLER, L.R. A heuristic algorithm for the vehicle dispatch problem. **Operations Research**, v.22, p.340-349, 1974.

GLOVER, F. Tabu Search-Part I, ORSA Journal on Computing. v.1, n.3 p.190-206, 1989.

GLOVER, F. Tabu Search-Part II, **ORSA Journal on Computing**, v.2, n.1, p.4-32, 1990.

GOLDBERG, D.E.; LINGLE, R. Alleles, loci and the traveling salesman problem. In: International conference on genetic algorithms, v.1, 1985, Mahwah, NJ. **Anais**... Mahwah: Lawrence Eribaum Associate, 1985, p.154-159.

GOLDEN, B.L.; WASIL, E.A.; KELLY, J.P; CHAO, I.M. The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results. In: Crainic TG, Laporte G, editors. **Fleet management and logistics**. Dordrecht: Kluwer, p. 33-56, 1998.

GONÇALVES, L.F.S. **Desenvolvimento de sistema de navegação autônoma por GNSS**. 2011. 192 p. Tese (Mestrado) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2011.

HO, S.C.; HAUGLAND, D. A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. **Computers & Operations Research**, v.31, n.12, p.1947-1964, 2004.

IOANNOU, G.; KRITIKOS, M.; PRASTACOS, G. A greedy look-ahead heuristic for the vehicle routing problem with time windows. **Operations Research Society**, v.52, p.523-537, 2001.

IRNICH, S. E VILLENEUVE, D. The shortest path problem with k-cycle elimination ($k \ge 3$): Improving a branch-and-price algorithm for the VRPTW. **INFORMS Journal of Computing**, Forthcoming, 2005.

KALLEHAUGE, B.; LARSEN, J. E MADSEN, O.B.G. Lagrangean duality and non-differentiable optimization applied on routing with time windows - experimental results. Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2001, (Internal report IMM-REP-2000-8)

KARGER, D.R., AND C. STEIN. An $\tilde{O}(n^2)$ algorithm for minimum cuts. In:ACM Symposium on the Theory of Computing, 25, 1993, San Diego. **Anais**... San Diego: ACM Press, 1993, p.757-765.

KIRKPATRICK, S. ET AL. Optimization by Simulated Annealing. **Science**, v.220, p.671-680, 1983.

KOHL, J., ET AL. 2-Path Cuts for the Vehicle Routing Problem with Time Windows, **Transportation Science**. v.33, n.1, p.101-116, 1999.

KELLY, J.P.; XU, J. A set-partitioning-based heuristic for the vehicle routing problem. **INFORMS Journal of Computing**, v.11, n.2, p.209-224, 1999.

KOLEN, A.W.J.; RINNOOY, A.H.G. E TRIENEKENS, H.W.J.M. Vehicle Routing with Time Windows. **Operations Research**, v.35, n.2, p.266-273, 1987.

KONTORAVDIS, G. A.; BARD,J. F. A GRASP for the vehicle routing problem with time windows. **INFORMS J. on Computing**, v.7, n.1, p.10-23, 1995.

LARSEN, J. **Parallelization of the vehicle routing problem with time windows**. 1999. 273 p. Tese (Ph.D) – Department of Mathematical Modelling, Technical University of Denmark, Lynghy, Denmark, 1999.

LINDEN, R. Algoritmos genéticos. Rio de Janeiro: Brasport, 2006. 372p.

NASSER A. EL-SHERBENY, Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. **Journal of King Saud University Science**, v.22, n.3, p.123-131, 2010.

NOVAES, A.G. Logística e gerenciamento da cadeia de distribuição 2ed. Rio de Janeiro: Campos, 2004. 424p.

OLIVEIRA, H.C.B., ET AL. A Robust Method for the VRPTW with Multi-Start Simulated Annealing and Statistical Analysis. In: IEEE Symposium Series on Computational Intelligence in Scheduling, 2007, Honolulu, USA. **Anais**... 2007, p.198-205.

OMBUKI, B.; ROSS, B.; HANSHAR, F. Multi-objective genetic algorithm for vehicle routing problem with time windows, **Applied Intelligence** v.24, p.17-30, 2006.

OSMAN, I. H. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problems. **Operations Research**, v.41, p.421-452, 1993.

POTVIN, J.Y.; ROUSSEAU, J.M. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. **European. J. Operations Research**, v.66, p.331-340, 1993.

POTVIN, J.Y. BENGIO, S. The Vehicle Routing Problem with Time Windows - Part II: Genetic Search. **Informs Journal on Computing**, v.8, p.165-172, 1996.

POTVIN, J.Y.; ROUSSEAU, J.M. An exchange heuristic for routing problems with time windows. **Journal of the Operational Research Society**, v.46, n.12, p.1433-1446, 1995.

PRINS, C. A simple and effective evolutionary algorithm for the vehicle routing problem. **Computers and Operations Research**, v.31, p.1985-2002, 2004.

ROCHAT, Y.; TAILLARD, E. D. Probabilistic diversification and intensification in local search for vehicle routing, **Journal of Heuristics**, v.1, p.147-167, 1995.

RONEN, D. Perspectives on practical aspects of truck routing and scheduling. **European Journal of Operational Research**, v.35, n.2, p.137-145, 1988.

RUSSELL, R.A. Hybrid heuristics for the vehicle routing problem with time windows. **Transportation Science**, v.29, n.2, p.156-166, 1995.

SHAW, P. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In: International Conference on Principles and Practice of Constraint Programming, 4, 1998, Pisa, Italy. **Anais**... London: Springer-Verlag, 1998, p.417-431.

SOLOMON, M.M. Algorithms for vehicle routing and scheduling problems with time window constraints. **Operations Research**, v.35, n.2, p.254-266, 1987.

SOLOMON, M. M.; DESROSIERS, J. Time Window Constrained Routing and Scheduling Problem. **Transportation Science**, v.22, n.1, p.1-13, 1988.

TAILLARD, É.D. Parallel iterative search methods for vehicle routing problems. **Networks**, v.23, n.8, p.661-673, 1993.

TAILLARD, É.D.; ET AL. A tabu search heuristic for the vehicle routing problem with soft time windows, **Transportation Science**, v.31, p.170-186, 1997.

TAN, K. C.; ET AL. Heuristic methods for vehicle routing problem with time windows. **Artificial Intelligence in Engineering**, v.15, p.281-295, 2001.

TAN, K. C.; CHEW, Y. H.; LEE, L. H. A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows. **Computational Optimization and Applications**, v.34, p.115-151, 2006.

THANGIAH, SAM R. Vehicle Routing with Time Windows using Genetic Algorithms. Submitted to the **International Journal on Computers and Operations Research**, 1993

THANGIAH, S.R.; OSMAN, I.H.; SUN, T. **Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows**. Computer Science Department, Slippery Rock University, 1994, (Technical Report 27).

THANGIAH, S. Vehicle routing with time windows using genetic algorithms, in: APPLICATIONS HANDBOOK OF GENETIC ALGORITHMS: NEW FRONTIERS, 1995, Boca Raton. Anais... Boca Raton: CRC Press, 1995, v.2, p.253-277.

THOMPSON, P. M.; PSARAFTIS, H.N. Cyclic transfer algorithms for multivehicle routing and scheduling problems. Operations Research, v.41, p.935-946, 1993.

URSANI, Z.; ET AL. Localized genetic algorithm for vehicle routing problem with time windows. **Applied Soft Computing**, v.11, n.8, p.5375-5390, 2011.

VIEIRA, H.P. Metaheurística para a solução de problemas de roteamento de veículos com janela de tempo. 2008. p.108, Tese (Mestrado) — Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas, Campinas, 2008.

WHYTE, H. D. A critical review of the state-of-the-art in autonomous land vehicle systems and technology. Springfield, 2001 (Sandia Report SAND2001-3685).