

INSTITUTO FEDERAL DO ESPÍRITO SANTO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE CONTROLE E
AUTOMAÇÃO

WARLEY ROCHA MENDES

**MODELO HÍBRIDO CONSTITUÍDO POR ALGORITMO GENÉTICO E *SPACE*
FILLING CURVE APLICADO À OTIMIZAÇÃO DE ROTAS VEICULARES**

SERRA
2017

WARLEY ROCHA MENDES

MODELO HÍBRIDO CONSTITUÍDO POR ALGORITMO GENÉTICO E *SPACE FILLING CURVE* APLICADO À OTIMIZAÇÃO DE ROTAS VEICULARES

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Controle e Automação do Instituto Federal do Espírito Santo, como requisito parcial para obtenção do título de Mestre em Engenharia de Controle de Automação.

Orientador: Prof. Dr. Daniel Cruz Cavalieri
Coorientador: Prof. Dr. Flávio Garcia Pereira

SERRA
2017

Dados Internacionais de Catalogação na Publicação (CIP)

M538m Mendes, Warley Rocha
2017 Modelo híbrido constituído por algoritmo genético e *Space Filling Curve* aplicado à otimização de rotas veiculares / Warley Rocha Mendes. - 2017.
139 f.; il.; 30 cm

Orientador: Prof. Dr. Daniel Cruz Cavalcante
Coorientador: Prof. Dr. Flávio Garcia Pereira.
Dissertação (mestrado) - Instituto Federal do Espírito Santo,
Programa de Pós-graduação em Engenharia de Controle de
Automação, 2017.

1. Sistema de informações geográficas. 2. Algoritmo genético.
3. Otimização matemática. Veículos. I. Cavalcante, Daniel Cruz. II.
Pereira, Flávio Garcia. III. Instituto Federal do Espírito Santo. IV.
Título.

CDD 629.8

Bibliotecária Rogeria Gomes Belchior - CRB6/ES 417

MINISTÉRIO DA EDUCAÇÃO
INSTITUTO FEDERAL DO ESPÍRITO SANTO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

WARLEY ROCHA MENDES

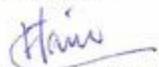
**MODELO HÍBRIDO CONSTITUÍDO POR ALGORITMO GENÉTICO E SPACE
FILLING CURVE APLICADO À OTIMIZAÇÃO DE ROTAS VEICULARES**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Controle e Automação do Instituto Federal do Espírito Santo, como requisito parcial para obtenção de título de Mestre em Engenharia de Controle e Automação.

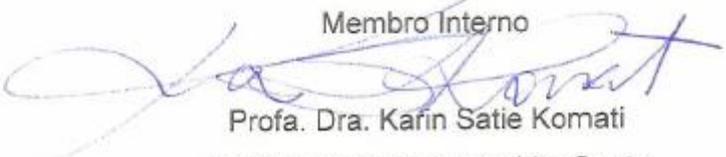
Aprovado em 12 de dezembro de 2017

COMISSÃO EXAMINADORA


Prof. Dr. Daniel Cruz Cavalieri
Instituto Federal do Espírito Santo
Orientador


Prof. Dr. Flávio Garcia Pereira
Instituto Federal do Espírito Santo
Orientador


Prof. Dr. Leandro Colombi Resendo
Instituto Federal do Espírito Santo
Membro Interno


Profa. Dra. Karin Satie Komati
Instituto Federal do Espírito Santo
Membro Interno


Prof. Dr. Jorge Leonid Aching Samatelo
Universidade Federal do Espírito Santo
Membro Externo

A todas as pessoas que amo e admiro.

Sem vocês nada disso faria sentido.

AGRADECIMENTOS

Primeiramente, agradeço a Deus por todas as conquistas alcançadas, pela vida, esperança, fé, coragem e por tudo que tenho e sou.

À minha amada esposa Marta Talitha Carvalho Freire Mendes, que sempre me apoiou de forma incondicional. Sua compreensão e colaboração foram fundamentais para a realização e conclusão deste trabalho.

À minha filha Sarah Freire Mendes pelo carinho, alegria, inspiração e razão de todo este esforço.

Ao Professor Dr. Daniel Cruz Cavalieri, orientador desta dissertação, o meu mais profundo respeito, amizade e gratidão. Muito obrigado pelos conselhos, orientações, incentivos e, principalmente, por acreditar em mim. Sem a sua participação seria impossível realizar esse trabalho.

Agradeço imensamente ao Professor Dr. Flávio Garcia Pereira, coorientador, pela disponibilidade, apoio, atenção prestadas, orientação e pelos importantes conselhos e ideias.

Também não posso deixar de agradecer aos meus pais Wanderlan e Maria Aparecida pelo amor incondicional, ensinamento e exemplo de vida. Aos meus irmãos Wallan, Kelly Cristiane e Wayne pelo carinho e união.

A todos os colegas e companheiros de jornada, em especial, Eduardo Lima Pereira e Ding Yih An, pelas constantes motivações, insistências e superações.

À Companhia Espírito Santense de Saneamento por todo apoio e investimento e ao Instituto Federal do Espírito Santo por ofertar o importante curso de Mestrado em Engenharia de Controle de Automação.

A todos vocês, **MUITO OBRIGADO!**

“Mas aqueles que esperam no Senhor renovam as suas forças.
Voam bem alto como águias; correm e não ficam exaustos,
andam e não se cansam”.

RESUMO

Alinhado à proposta do Mestrado Profissional, este trabalho é aplicado na resolução de um problema real de roteamento existente na Companhia Espírito Santense de Saneamento, que hoje dispõe de um Sistema de Informações Geográficas, mas não possui nenhum módulo ou mecanismo de geração de rotas de atendimento aos seus clientes, os quais abrem, diariamente, em média 2148 solicitações de serviços. Nesse contexto, este trabalho apresenta a proposta de um Algoritmo Híbrido de Otimização, formado pela combinação dos Algoritmos Genéticos e *Space-Filling Curves*, aplicado a resolução do Problema de Roteamento de Veículos. Além da elaboração de um módulo de geração de rotas automáticas utilizando informações georreferenciadas existente no sistema. Ao longo deste trabalho também é demonstrado a viabilidade do algoritmo híbrido proposto por meio de testes realizados em duas bases distintas de *benchmark* existentes na literatura. Comparações realizadas demonstram que o algoritmo chegou a alcançar um resultado médio de 12,7% em relação a solução ótima da primeira base e de 4,1% da segunda, em relação a solução ótima. Em complemento, com intuito de reforçar a validação do algoritmo proposto, foram realizadas comparações entre ele e cinco variações do Algoritmo da Otimização da Colônia de Formigas. Nessa comparação, os resultados alcançados demonstram a superioridade do algoritmo híbrido em quase todas as simulações, comprovando a viabilidade do método proposto.

Palavras-chave: Roteamento de veículos. Algoritmo genético. Curvas de preenchimentos de espaços. Algoritmo híbrido de otimização.

ABSTRACT

Aligned to the proposal of the Professional Master's Degree, this work is applied in the resolution of a real routing problem in the Espírito Santense de Saneamento Company, which now has a Geographic Information System, but does not have any module or mechanism to generate service routes to its clients, who daily open an average of 2148 requests for services. In this context, this work presents the proposal of a Hybrid Optimization Algorithm, formed by the combination of Genetic Algorithms and Space-Filling Curves, applied to the resolution of the Vehicle Routing Problem. In addition to the elaboration of an automatic route generation module using existing georeferenced information in the system. Throughout this work the feasibility of the proposed hybrid algorithm is also demonstrated through tests carried out in two benchmark databases in the literature. Comparisons showed that the algorithm reached an average result of 12.7% in relation to the optimal solution of the first base and of 4.1% of the second one, in relation to the optimal solution. In addition, in order to reinforce the validation of the proposed algorithm, comparisons were made between it and five variations of the Antimony Optimization Algorithm. In this comparison, the obtained results demonstrate the superiority of the hybrid algorithm in almost all the simulations, proving the feasibility of the proposed method.

Keywords: Vehicle routing. Genetics algorithm. Space-Filling Curve. Hybrid optimization algorithm.

LISTA DE FIGURAS

Figura 1 - Etapas do planejamento do desenvolvimento da pesquisa.	19
Figura 2 - Representação de um exemplo do CVRP usando um depósito de distribuição central.....	30
Figura 3 - Estrutura dos Métodos de Otimização Clássica.....	39
Figura 4 - Fluxograma do AG tradicional.....	55
Figura 5 - Exemplo de seleção por roleta.....	59
Figura 6 - Exemplo de cruzamento de um ponto.....	62
Figura 7 - Exemplo de cruzamento de dois pontos.	62
Figura 8 - Exemplo de cruzamento uniforme.....	63
Figura 9 - Exemplo de cruzamento ordenado.	64
Figura 10 - Exemplo do operador de mutação.	65
Figura 11 - Exemplo de <i>Exchange Mutation</i>	67
Figura 12 - Mapas das cidades em um problema do caixeiro viajante.....	68
Figura 13 - Representação de indivíduo contendo uma solução parcial.	69
Figura 14 - Representação da solução parcial do problema.	70
Figura 15 - Melhor solução encontrada para o problema.....	71
Figura 16 - Mapeamento e curva de Peano.	74
Figura 17 - Curva de Peano com 6 divisões por lado.....	74
Figura 18 - Curvas de Peano com 9 e 27 divisões nas faces.	75
Figura 19 - Curva de Hilbert em diferentes ordens.....	76
Figura 20 - Exemplo de decomposição da Curva de Hilbert.	76
Figura 21 - Sequência da ordem de mapeamento da Curva de Hilbert.	77
Figura 22 - Regras de subdivisão dos copos existentes nas curvas de Hilbert.....	78
Figura 23 - Regra de decomposição de níveis.	79
Figura 24 - Níveis de iteração da função de geração das curvas de Hilbert.	80
Figura 25 - Pontos georreferenciados no sistema GIS.....	82
Figura 26 - Pontos encaixados na matriz de mapeamento da Curva de Hilbert de ordem 6.	82
Figura 27 - Rota gerada por meio da sequência da ordem de mapeamento da Curva de Hilbert.	83
Figura 28 - Médias dos resultados alcançados pelo AG em sete instâncias contidas na base de <i>benchmark</i>	95
Figura 29 - Exemplo de cruzamento totalmente aleatório.	96

Figura 30 - Comparação dos resultados entre os cruzamentos ordenado e totalmente aleatório.....	97
Figura 31 - Operador de Rearranjo Genético.....	98
Figura 32 - Processo de deslocamento do gene da esquerda para a direita.	99
Figura 33 - Processo de deslocamento do gene da direita para a esquerda.	99
Figura 34 - Estrutura do AG com o operador de Rearranjo Genético.	100
Figura 35 - Representação da mutação por <i>Exchange Mutation</i> após alteração...	101
Figura 36 - Comparação do resultado da solução ao adicionar às melhorias ao AG.....	102
Figura 37 - Representação do grafo após a divisão da área de busca pela Curva de Hilbert de ordem 1.....	105
Figura 38 - Mapeamento fractal de cada agrupamento da área de busca.	105
Figura 39 - Calibração do parâmetro do número de indivíduos da população do algoritmo híbrido.....	107
Figura 40 - Configuração do parâmetro da taxa de mutação do algoritmo híbrido..	107
Figura 41 - Calibração do critério de parada do algoritmo híbrido.....	108
Figura 42 - Resultado do algoritmo após a etapa de calibração.	109
Figura 43 - Instância P-n101-k4 da base de <i>benchmark</i> , espaço de busca contendo 4 veículos e 100 clientes.	111
Figura 44 - Solução ótima contendo custo de 681.	111
Figura 45 - Resultado do algoritmo híbrido contendo custo de 754, uma diferença de 10,7% da solução ótima.	112
Figura 46 - Comparação entre a variação dos custos das rotas.	115
Figura 47 - Instância eil51 da base de <i>benchmark</i> , contendo 7 veículos e 50 clientes.....	116
Figura 48 - Resultado do algoritmo híbrido proposto com custo final de 631,59, uma diferença de 6,2% da solução ótima.	116
Figura 49 - Instância rat99 da base de <i>benchmark</i> , contendo 7 veículos e 98 clientes.....	117
Figura 50 - Resultado do algoritmo híbrido proposto com custo final de 2703,38, uma diferença de 18,0% da solução ótima.	117
Figura 51 - Mapa do sistema GIS contendo 256 pontos de Solicitação de Serviços georreferenciados.....	119
Figura 52 - Rota gera pelo algoritmo híbrido proposto de custo final 17.883 metros.	119
Figura 53 - Rota gera pelo algoritmo híbrido proposto com custo final de 19.477 metros.	120

LISTA DE TABELAS

Tabela 1 - Características da metodologia científica aplicada ao trabalho.....	18
Tabela 2 - Dimensão do espaço de busca em função do número de cidades do TSP.	27
Tabela 3 - Pseudocódigo do Algoritmo de Genético.	55
Tabela 4 - Pseudocódigo do método de seleção por roleta.	60
Tabela 5 - Pseudocódigo do método de mutação uniforme.	66
Tabela 6 - Pseudocódigo do método de mutação ordenada.	66
Tabela 7 - Distâncias entre as cidades.	69
Tabela 8 - Código de implementação das curvas de Hilbert.	79
Tabela 9 - Comparação dos resultados alcançados pelo AG em relação como os resultados ótimos da base de <i>benchmark</i>	94
Tabela 10 - Comparação dos resultados alcançados pelo AG contendo melhorias e sua implementação padrão.	102
Tabela 11 - Comparação entre todas as abordagens experimentadas.	110
Tabela 12 - Comparação do tempo de execução entre o CPLEX e o Algoritmo Híbrido.	114
Tabela 13 - Comparação dos resultados alcançados por cada algoritmo.	114

LISTA DE ABREVIATURAS E SIGLAS

AG – Algoritmo Genético

API – *Application Programming Interface*

AOC – Algoritmo da Otimização da Colônia de Formigas

CESAN – Companhia Espírito Santense de Saneamento

CVRP – *Capacitated Vehicle Routing Problem*

GVRP – *Green Vehicle Routing Problem*

HOA – *Hybrid Optimization Algorithm*

OVRP – *Open Vehicle Routing Problem*

MDVRP – *Multiple Depots Vehicle Routing Problem*

MTSP – *Multiple Traveling Salesman Problem*

PDVRP – *Pickup-and-Delivery Vehicle Routing Problem*

SDVRP – *Single Depot Vehicle Routing Problem*

SFC – *Space-filling Curve*

SVRP – *Stochastic Vehicle Routing Problem*

TSP – *Traveling Salesman Problem*

VRP – *Vehicle Routing Problem*

VRPTW – *VRP with Time Windows*

SUMÁRIO

1	INTRODUÇÃO	15
1.1	CONTEXTUALIZAÇÃO	15
1.2	MOTIVAÇÃO	16
1.3	OBJETIVOS	16
1.4	CONTRIBUIÇÕES.....	17
1.5	METODOLOGIA DE DESENVOLVIMENTO DO TRABALHO	17
1.6	TRABALHOS CORRELATOS	19
1.7	QUESTÕES NORTEADORAS.....	21
1.8	ESTRUTURA DA DISSERTAÇÃO	21
1.9	PUBLICAÇÕES	22
2	PROBLEMAS DE ROTEIRIZAÇÃO DE VEÍCULOS	24
2.1	TIPOS DE PROBLEMAS DE ROTEIRIZAÇÃO DE VEÍCULOS.....	25
2.1.1	Problema do caixeiro viajante (<i>Traveling Salesman Problem - TSP</i>)	25
2.1.2	Problema de múltiplos caixeiros-viajantes (<i>Multiple Traveling Salesman Problem - MTSP</i>)	27
2.1.3	<i>Single Depot VRP (SDVRP)</i>	28
2.1.4	<i>Multiple Depots VRP (MDVRP)</i>	28
2.1.5	<i>Capacitated Vehicle Routing Problem (CVRP)</i>	28
2.1.6	<i>Open Vehicle Routing Problem (OVRP)</i>	30
2.1.7	<i>VRP with Time Windows (VRPTW)</i>	31
2.1.8	<i>Pickup-and-Delivery VRP (PDVRP)</i>	32
2.1.9	<i>Stochastic VRP (SVRP)</i>	32
2.1.10	<i>Green VRP (GVRP)</i>	33
2.2	PRINCIPAIS RESTRIÇÕES EXISTENTES EM VRPS.....	33
2.3	METODOLOGIAS DE RESOLUÇÃO PARA O VRP	39
2.3.1	Métodos exatos	40
2.3.2	Métodos aproximados ou heurísticos	40
2.3.3	Métodos heurísticos e meta-heurísticos	41
2.4	PARÂMETROS DOS MÉTODOS DE OTIMIZAÇÃO	42
3	ALGORITMOS GENÉTICOS E <i>SPACE-FILLING CURVES</i>	48
3.1	ALGORITMOS GENÉTICOS	48
3.1.1	Características gerais dos algoritmos genéticos	49

3.1.2	Métodos e critérios para a implementação do algoritmo genético	53
3.1.3	Seleção de indivíduos	56
3.1.3.1	Seleção aleatória.....	56
3.1.3.2	Seleção por elitismo	57
3.1.3.3	Seleção por roleta	58
3.1.3.4	Seleção por torneio	60
3.1.4	Cruzamento (<i>Crossover</i>)	60
3.1.4.1	Cruzamento de um ponto.....	61
3.1.4.2	Cruzamento de dois pontos.....	62
3.1.4.3	Cruzamento uniforme	63
3.1.4.4	Cruzamento ordenado (<i>Order Crossover - OX</i>).....	63
3.1.5	Mutação	64
3.1.5.1	<i>Exchange Mutation</i>	66
3.1.6	Critério de parada do algoritmo genético	67
3.1.7	Aplicação do Algoritmo genético em problemas de rota	67
3.1.8	Pontos positivos na utilização do AG na resolução de problemas de geração de rotas automáticas	71
3.1.9	Pontos negativos na utilização do AG na resolução de problemas de geração de rotas automáticas	72
3.2	<i>SPACE-FILLING CURVES</i>	73
3.2.1	Curvas de Peano	73
3.2.2	Mapeamento da Curva de Hilbert	76
3.2.3	Implementação do algoritmo das Curvas de Hilbert	77
3.2.4	Aplicação da Curva de Hilbert em problemas de rota	80
3.2.5	Pontos negativos na utilização da Curva de Hilbert na resolução de problemas de geração de rotas automáticas	83
3.2.6	Pontos positivos na utilização da Curva de Hilbert na resolução de problemas de geração de rotas automáticas	84
4	RESTRIÇÕES, CATEGORIZAÇÃO E MODELO MATEMÁTICO	85
4.1	CRITÉRIOS DO PROBLEMA PROPOSTO	85
4.2	CATEGORIZAÇÃO DO PROBLEMA DE ROTA	86
4.2.1	Tipo de problema de rota	87
4.3	FORMULAÇÃO MATEMÁTICA.....	87
5	ALGORITMO HÍBRIDO PROPOSTO, RESULTADOS, COMPARAÇÕES E DISCUSSÕES	92
5.1	APLICAÇÃO DO AG NA RESOLUÇÃO DE UM PROBLEMA PROPOSTO .	92
5.1.1	Cruzamento totalmente aleatório (<i>Random Crossover</i>)	95

5.1.2	Operação de rearranjo genético	97
5.2	ALGORITMO HÍBRIDO PROPOSTO.....	102
5.3	RESULTADOS E COMPARAÇÕES.....	108
6	CONCLUSÃO	121
6.1	TRABALHOS FUTUROS	122
	REFERÊNCIAS	123
	APÊNDICE A - COMPARAÇÕES DOS RESULTADOS ALCANÇADOS....	123
	APÊNDICE B - GRAFOS	123

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Ao longo de sua história a Companhia Espírito Santense de Saneamento (CESAN) vem tentando resolver alguns de seus maiores desafios: coletar amostras de águas em determinados pontos existentes nas 52 cidades atendidas pela empresa com o intuito de verificar a qualidade em que a água está chegando às casas, comércios e empresas, realizar vistorias e inspeções em hidrantes, bueiros, elevatórias e registros, para manter a usabilidade e funcionamento desses elementos, e também fiscalizar denúncias e possíveis clientes clandestinos existentes em localidades espalhadas ao longo das cidades. Tudo isso dentro de prazos legais estabelecidos pelos órgãos fiscalizadores (AGÊNCIA DE REGULAÇÃO DE SERVIÇOS PÚBLICOS DO ESPÍRITO SANTO, 2017) e, principalmente, para manter a excelência dos serviços prestados pela empresa e manter a satisfação de seus clientes.

Em todos os casos citados anteriormente é necessário que profissionais da empresa se desloquem para verificar e investigar as diversas solicitações realizadas pelos clientes, abertas no meio de canais de comunicação existente na empresa (telefone, site e dispositivos móveis). Por dia, é aberta uma média de 2148 solicitações de serviços, dessas, 800 somente da grande Vitória.

Para realizar todos esses trabalhos é de extrema importância que seja criado o plano de rota que guiará, em tempo hábil, as equipes de atendimento externo aos locais onde serão realizadas as suas atividades ao longo do dia.

No final de 2014, a CESAN implantou em suas dependências um sistema corporativo de informações geográfica (SIG ou GIS - *Geographic Information System*) que permitiu representar no espaço geográfico diferentes tipos de informações existentes em seu domínio (FITZ, 2008).

Atualmente, com base no posicionamento geográfico das ocorrências e com o auxílio de um conjunto de camadas cartográficas existentes no sistema GIS, os planejadores criam diariamente diferentes conjuntos de rotas a serem visitados de forma manual, baseando-se somente em seus conhecimentos, experiências e na disposição das informações apresentadas nos mapas.

1.2 MOTIVAÇÃO

Pelo fato da CESAN possuir um sistema corporativo de informações geográficas contendo as diversas solicitações dos clientes georreferenciadas, e que a execução das solicitações é planejada de forma manual, demandando um número elevado de tempo e de pessoas, vislumbrou-se a oportunidade de automatizar e tornar eficiente o processo de planejamento de rotas.

1.3 OBJETIVOS

Dado o cenário motivacional apresentado acima, o objetivo principal desse trabalho é utilizar um algoritmo de percurso em grafo baseado na teoria de *Space-Filling Curves* combinada com Algoritmo Genético, visando possibilitar a geração automática de rotas de menor custo para diferentes camadas de informações geográficas, sem que haja intervenção humana. Além disso, os seguintes objetivos específicos foram estabelecidos:

- Validar o algoritmo proposto por meio da comparação dos seus resultados alcançados com a solução ótima em duas bases de *benchmarks* distintas;

- Realizar comparações entre o algoritmo híbrido proposto com outra heurística, o Algoritmo de Colônia de Formigas, com objetivo de comparar os resultados alcançados por cada abordagem;
- Implementar um novo módulo no sistema GIS utilizando mapas temáticos de diferentes pontos georreferenciados localizados nos limites do estado do Espírito Santo.

1.4 CONTRIBUIÇÕES

Como contribuição científica, desenvolver um Algoritmo Híbrido de Otimização (*Hybrid Optimization Algorithm - HOA*), formado pela combinação do Algoritmo Genético e do Algoritmo de *Space-Filling Curves* aplicado à resolução de problemas de roteirização de veículos usando informações georreferenciadas.

Como contribuição prática, alinhada à proposta do Mestrado Profissional em Engenharia de Controle e Automação, o algoritmo proposto será utilizado na resolução do problema de roteamento de veículos da CESAN, que atualmente realiza manualmente toda as atividades de planejamento de rotas veiculares de atendimento aos seus clientes de forma manual, o que demanda uma quantidade elevada de tempo e pessoas, além de gerar soluções que não possuem nenhuma garantia de qualidade.

1.5 METODOLOGIA DE DESENVOLVIMENTO DO TRABALHO

Pode-se definir pesquisa como o procedimento racional e sistemático que tem como objetivo proporcionar respostas aos problemas que são propostos. A pesquisa é requerida quando não se dispõe de informação suficiente para responder ao

problema, ou então quando a informação disponível se encontra em tal estado de desordem que não passa ser adequadamente relacionada ao problema (GIL, 2002).

A pesquisa é desenvolvida mediante o conjunto de conhecimento disponível e a utilização de métodos, técnicas e outros procedimentos científicos. Na realidade, a pesquisa desenvolve-se ao longo do processo que envolve inúmeras fases, desde a adequada formulação do problema até a satisfatória apresentação dos resultados (KAHLMAYER-MERTENS et al., 2007).

Não há, evidentemente, regras fixas acerca da elaboração de um projeto. Sua estrutura é determinada pelo tipo de problema a ser pesquisado e também pelo estilo de seus autores. É necessário que o projeto esclareça como se processará a pesquisa, quais as etapas que serão desenvolvidas e quais os recursos que devem ser alocados para atingir seus objetivos (GIL, 2002).

A principal razão que determina a realização desta pesquisa pode ser classificada como de ordem prática, pois decorre do desejo de conhecer e realizar algo mais eficiente ou eficaz para ser posteriormente utilizado pela CESAN.

A pesquisa realizada no presente trabalho enquadra-se no campo das pesquisas Teórico-Aplicada Descritiva. Em sentido amplo de pesquisa, a investigação científica tratada neste trabalho tem por metodologia as seguintes características:

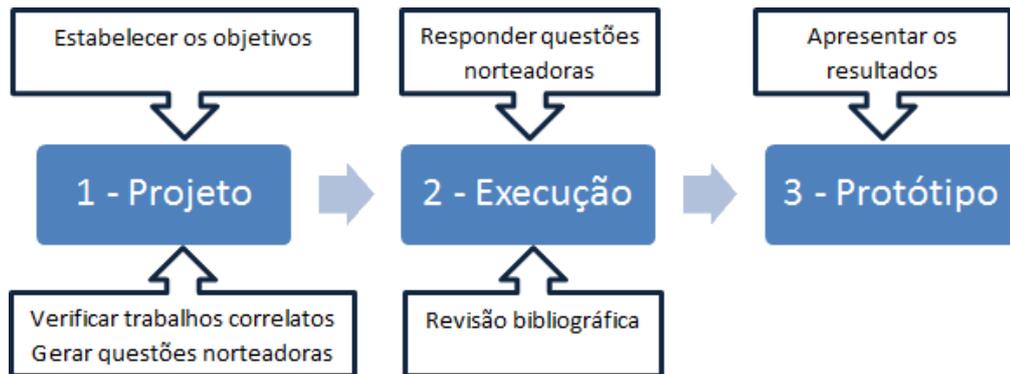
Tabela 1 - Características da metodologia científica aplicada ao trabalho.

Quanto à natureza	Pesquisa aplicada
Quanto aos objetivos	Exploratório e descritiva
Quanto às abordagens	Quantitativo
Quanto aos procedimentos	Experimental

Fonte: Próprio autor.

Baseado nestas características o trabalho foi desenvolvido conforme as etapas apresentadas na Figura 1. Essas etapas englobam fases, processos, ações principais e produção de artefatos.

Figura 1 - Etapas do planejamento do desenvolvimento da pesquisa.



Fonte: Próprio autor.

A primeira fase, “Projeto”, preocupa-se em formar os limites da pesquisa e definir os objetivos. Na fase seguinte, “Execução”, são realizadas a pesquisa bibliográfica, análises, interpretações, e que por meio de uma reflexão crítica as questões norteadoras são clareadas e o projeto é construído. Por final o protótipo é finalizado e os resultados são apresentados.

1.6 TRABALHOS CORRELATOS

Há mais de uma década pesquisadores como (RIBEIRO; LORENA, 2005), (WU; CUNHA, 2008) e (REBELLO; HAMACHER, 2000) vem apresentando propostas consistentes na otimização de rotas aplicadas ao problema de roteamento de veículos (*Vehicle Routing Problem - VRP*) baseando-se em Algoritmos Genéticos e suas variações. Sendo um destaque para (HEINEN; OSÓRIO, 2006), que realizou em seu trabalho um importante comparativo entre três heurísticas de aproximação: a

heurística Clark e Wright (CLARKE; WRIGHT, 1964), a heurística de Mole e Jameson (1976) e os Algoritmos Genéticos (GOLDBERG, 1989). Nesse trabalho os autores realizam diversos experimentos para cada heurística e demonstram comparações relativas ao tempo de execução e a qualidade das soluções encontradas. Em sua conclusão, eles destacam os resultados positivos alcançados pelo Algoritmo Genético e a superioridade em relação às outras duas heurísticas.

Em trabalhos mais recentes, publicações demonstram ainda a forte utilização e recomendação do Algoritmo Genético na resolução de problemas de roteamento de veículos, como o apresentado por (VIDAL, 2013a) que afirma que “os Algoritmos Genéticos podem resolver uma larga variedade de VRPs e suas variações”.

Em outro trabalho, os autores Habibeh Nazif e Lai SoonLee (2012) propõem um algoritmo que utiliza um operador de cruzamento otimizado, projetado para um grafo bipartido completo não direcionado para encontrar um conjunto ótimo de rotas de entrega que satisfaçam os requisitos e proporcionam um custo total mínimo para problemas de roteamento de veículo capacitados (CVPR). Segundo os autores, o algoritmo foi testado com exemplos de *benchmark* e comparado com outras heurísticas existentes na literatura. Os resultados computacionais obtidos mostraram que o algoritmo proposto é competitivo em termos de qualidade das soluções encontradas.

Utilizando uma metodologia diferente das convencionais existentes na literatura, Bartholdi e Platzman demonstraram pela primeira vez por meio de seus artigos em Bartholdi III (1988, 1989); Platzman e Bartholdi III (1983) uma solução que consiste em uma programação dinâmica simples para a resolução do problema do caixeiro viajante utilizando *Space-Filling Curves*. Ele afirma que para construir um curto percurso nos pontos no plano, os pontos são sequenciados à medida que aparecem ao longo de uma curva de preenchimento de espaço. Esta heurística consiste essencialmente na triagem, por isso, é facilmente codificada e requer apenas $O(N)$ de memória e $O(N \log N)$ de operações. Seu desempenho é competitivo com o de outros métodos rápidos.

Em seguida, Bartholdi evoluiu os seus trabalhos realizando novas publicações relacionadas ao tema de otimização de rotas e apresentou à comunidade científica uma nova proposta de utilização com *Space-Filling Curves* aplicada à solução de problemas de roteamento de veículos, como demonstrados em Bartholdi et al. (1983), Bartholdi III (2017a, 2017b) e Bartholdi e Goldsman (2001a, 2001b), nos quais apresentou resultados positivos e a viabilidade ao utilizar essa abordagem, o que inspirou e motivou a utilização de duas técnicas na resolução do problema de geração automática de rotas em um sistema GIS, tema deste trabalho.

1.7 QUESTÕES NORTEADORAS

- O Algoritmo Genético e o Algoritmo *Space-Filling Curves* podem ser combinados possibilitando a criação de um algoritmo de otimização híbrido?
- Esse algoritmo de otimização híbrido pode ser aplicado na resolução de problemas de roteirização veicular?
- Os resultados alcançados pelo algoritmo híbrido serão superiores aos alcançados pela implementação padrão do Algoritmo Genético?
- Esse algoritmo de otimização híbrido pode ser implementado no sistema de informações geográficas da CESAN para a geração automática de rotas, levando em consideração as restrições do problema?

1.8 ESTRUTURA DA DISSERTAÇÃO

A presente Dissertação de Mestrado encontra-se dividida da seguinte forma:

- No Capítulo 1, é definido o contexto do projeto de forma introdutória, apresentando uma visão geral. Salientamos a motivação, objetivo, metodologia, trabalhos correlatos e as questões norteadoras.
- No Capítulo 2, são apresentados os tipos de problema de roteamento de veículos, as principais restrições existentes em roteamento, as metodologias de resolução, os parâmetros dos métodos de otimização e uma contextualização sobre grafos.
- O Capítulo 3 apresenta os Algoritmos Genético e de *Space-Filling Curves*, demonstra a aplicação desses algoritmos na geração de rotas e lista seus pontos positivos e negativos.
- Já o Capítulo 4 apresenta as restrições do problema de rotas tratado neste trabalho, sua categorização e o formalismo matemático do problema em questão.
- No Capítulo 5, é apresentado o Algoritmo Híbrido Proposto, as etapas de calibração do algoritmo, validação e seus resultados alcançados. É utilizada duas bases distintas de *benchmark* para a realização de comparações entre o algoritmo proposto e as soluções ótimas e também para a comparações entre o algoritmo híbrido proposto e cinco variações do algoritmo de Colônia de Formigas.
- Por fim, no Capítulo 6, são apresentadas as conclusões finais, bem como propostas para continuidade desta pesquisa.

1.9 PUBLICAÇÕES

Os esforços durante o processo de pesquisa para esta dissertação possibilitaram as seguintes publicações:

- MENDES, W. R.; Cavalieri, D. C.; Pereira, F. G. Utilização de Space-Filling Curves aplicado à otimização de rotas veiculares de menor custo com desvio de pontos de intervenção. In: SBAI Simpósio Brasileiro de Automação Inteligente, 2017, Porto Alegre - RS.
- MENDES, W. R.; Cavalieri, D. C.; Pereira, F. G.; Cuadros, M. A. S. L.; Almeida, G. M. Aplicação do algoritmo A* em um Sistema de Informações Geográficas para a identificação da melhor rota. In: ENCOSIS Encontro Regional de Computação e Sistemas de Informação, 2016, Manaus. Anais do Encontro Regional de Computação e Sistemas de Informação, 2016. v. 1. p. 1-221.

2 PROBLEMAS DE ROTEIRIZAÇÃO DE VEÍCULOS

O transporte rodoviário é o meio predominante de transportar mercadorias no Brasil e em outras partes do mundo. Os custos diretos associados a este tipo de transporte aumentaram significativamente devido, principalmente, ao aumento dos preços do combustível. Além disso, o transporte rodoviário está intrinsecamente associado a grandes quantidades de custos indiretos ou externos, que normalmente são facilmente observáveis, como congestionamento, vias esburacadas, custos relacionados à insegurança, mobilidade, etc (RIBEIRO; LORENA, 2005).

Nesse cenário, torna-se evidente que novos métodos devem ser desenvolvidos para suportar o processo de tomada de decisão para que as estratégias ótimas (ou quase ótimas) possam ser escolhidas no transporte rodoviário. E essa necessidade de otimizar o transporte rodoviário afeta os setores público e privado e constitui um grande desafio para a maioria das regiões industrializadas (CACERES-CRUZ et al., 2015).

Dantzig e Ramser (1959) foram os primeiros pesquisadores a introduzir o problema de despacho de caminhão ("*Truck Dispatching Problem*"), modelando como uma frota de caminhões homogêneos poderia atender à demanda de combustível de uma série de postos de gasolina a partir de uma central de distribuição com uma distância mínima a ser percorrida.

Cinco anos depois, Clarke e Wright (1964) generalizaram esse problema para um problema de otimização linear comumente encontrado no domínio da logística de transporte, cujo problema inicial era como servir um conjunto de clientes, geograficamente dispersos em torno de um depósito central, utilizando uma frota de caminhões com capacidades variáveis. Esse estudo ficou mundialmente conhecido como Problema de Roteirização de Veículo (*Vehicle Routing Problem - VRP*), sendo atualmente um dos tópicos mais amplamente estudados no campo da Pesquisa Operacional.

Os modelos atuais de VRP, no entanto, são imensamente diferentes dos introduzidos inicialmente, pois eles visam cada vez mais incorporar complexidades e restrições da vida real, como o tempo de deslocamento entre diferentes clientes, janelas de tempo para recolha e entrega de produtos, e informações de entrada (por exemplo, informações de demanda) que mudam dinamicamente ao longo do tempo, cujas características trazem uma complexidade substancial ao problema (BRAEKERS; RAMAEKERS, VAN NIEUWENHUYSE, 2016).

Nesse contexto, o objetivo principal do chamado Problema de Roteirização de Veículo é otimizar o *design* de roteamento (processo de distribuição de depósitos para clientes) de tal forma que a demanda de produtos dos clientes seja satisfeita sem violar qualquer restrição específica do problema - por exemplo, distância máxima da rota ou restrições relacionadas ao tempo (GOLDEN; RAGHAVAN; WASIL, 2008).

2.1 TIPOS DE PROBLEMAS DE ROTEIRIZAÇÃO DE VEÍCULOS

O VRP possui muitas variantes dependendo dos parâmetros, restrições e características consideradas. Nas próximas seções é apresentada uma classificação das principais variações desse modelo.

2.1.1 Problema do caixeiro viajante (*Traveling Salesman Problem - TSP*)

O problema do caixeiro viajante (TSP) é um dos mais tradicionais e conhecidos problemas de programação matemática. Os problemas de roteamento lidam em sua maior parte com passeios ou tours sobre pontos de demandas ou ofertas. Esses pontos podem ser representados por cidades, postos de trabalho ou atendimento, depósitos, clientes, etc. Dentre os tipos de passeios um dos mais importantes é o

denominado circuito hamiltoniano, criado por Willian Rowan Hamilton, que propôs um jogo de nome *Around the World* cujo desafio consistia em encontrar uma rota através dos vértices de um dodecaedro de tal modo que a rota iniciasse e terminasse no mesmo vértice, sem nunca repetir uma visita. Assim, o objetivo do TSP é encontrar em um grafo $G = (N, A)$ o caminho hamiltoniano de menor custo, de forma que todos os vértices sejam visitados uma única vez (GOLDBARG; LUNA, 2005).

No problema proposto consiste em determinar um único roteiro com menor custo possível que permita o caixeiro-viajante (veículo) visitar todos os nós (clientes) de uma rede uma única vez. O problema é baseado em um único depósito e o veículo deve sair e retornar à mesma base. Nesse problema não há restrição de capacidade de veículo, e a demanda é determinística.

O Problema do Caixeiro Viajante pode apresentar caminhos simétricos (quando a distância na ida e na volta entre duas cidades é a mesma) ou caminhos assimétricos (distâncias diferentes). A versão assimétrica apresenta maior grau de realidade, pois no trânsito das cidades muitas vezes não podemos utilizar o mesmo caminho na ida e na volta, tendo, portanto, distâncias diferentes. Ao contrário do que possa parecer, a versão simétrica é mais difícil de ser resolvido, pois os caminhos de distâncias idênticas criam ciclos que precisam ser evitados.

A maior dificuldade na resolução do TSP é quando há um número muito grande de possíveis rotas para n cidades, que é calculada por meio da Equação 2.1 para problema representado por um grafo completo (MALAQUIAS *et al.*, 2006). A Tabela 2.1 apresenta o número de rotas possíveis em função do número de cidades que possa vir a existir no espaço de busca do problema. Observa-se que o número de rotas cresce exponencialmente com número de cidades, o que inviabiliza a possibilidade de utilização de técnicas exatas de busca para encontrar a melhor solução.

$$\frac{(n-1)!}{2} \quad (2.1)$$

Tabela 2 - Dimensão do espaço de busca em função do número de cidades do TSP.

Número de Cidades	Número de Rotas
3	1
4	3
5	12
6	60
7	360
8	2.520
9	20.160
10	181.440
11	1.814.400
12	19.958.400
13	239.500.800
14	3.113.510.400
15	43.589.145.600
16	653.837.184.000

Fonte: Próprio autor.

2.1.2 Problema de múltiplos caixeiros-viajantes (*Multiple Traveling Salesman Problem - MTSP*)

O problema de múltiplos caixeiros-viajantes (MTSP) é uma extensão do problema do caixeiro-viajante, porém, ao invés de uma única rota, determinam-se múltiplos roteiros. O problema consiste em determinar múltiplos rotas com o menor custo possível, de modo que cada caixeiro-viajante deva visitar pelo menos um nó da rede, e cada nó deve ser visitado uma única vez. O problema é baseado em um único depósito e o veículo deve retornar e sair da mesma base. Nesse problema não há restrições de capacidade de veículo e a demanda é determinística (GOLDBARG; LUNA, 2005).

2.1.3 *Single Depot VRP (SDVRP)*

Esse é tipo de problema de VRP com apenas um depósito e nenhuma limitação de capacidade dos veículos utilizados nas rotas. Ele também é denominado por alguns autores de problema de roteamento de veículo de depósito único, que representa o mesmo problema dos múltiplos caixeiros-viajantes, sendo, neste caso, contextualizado no universo dos problemas de rotas veiculares (KARAKATIČ; PODGORELEC, 2015).

2.1.4 *Multiple Depots VRP (MDVRP)*

O problema de roteirização de veículos com múltiplos depósitos é uma generalização do problema clássico de roteirização, em que, ao invés de existir um único depósito, uma empresa possui vários, que estão distribuídos geograficamente, dos quais ela pode servir seus clientes. Dessa forma, algumas rotas são formadas por pontos de partida que iniciam em um determinado depósito e terminam em algum outro diferente (MONTROYA-TORRES et al., 2016).

2.1.5 *Capacitated Vehicle Routing Problem (CVRP)*

Esse problema é uma extensão do problema de múltiplos caixeiros-viajantes, em que se acrescenta a restrição de capacidade de veículos e tem como objetivo encontrar um conjunto de rotas de menor custo possível, iniciando e terminando no depósito, sendo que, cada veículo da frota possui capacidades e características similares e cada cliente possui certa demanda que deve ser satisfeita. Além disso, há uma matriz de custos que mede os custos associados à movimentação de um

veículo de um nó para outro. Esses custos geralmente representam distâncias, tempos de viagem, número de veículos empregados ou uma combinação desses fatores (CACERES-CRUZ et al., 2015) (BRAEKERS; RAMAEKERS, VAN NIEUWENHUYSE, 2016).

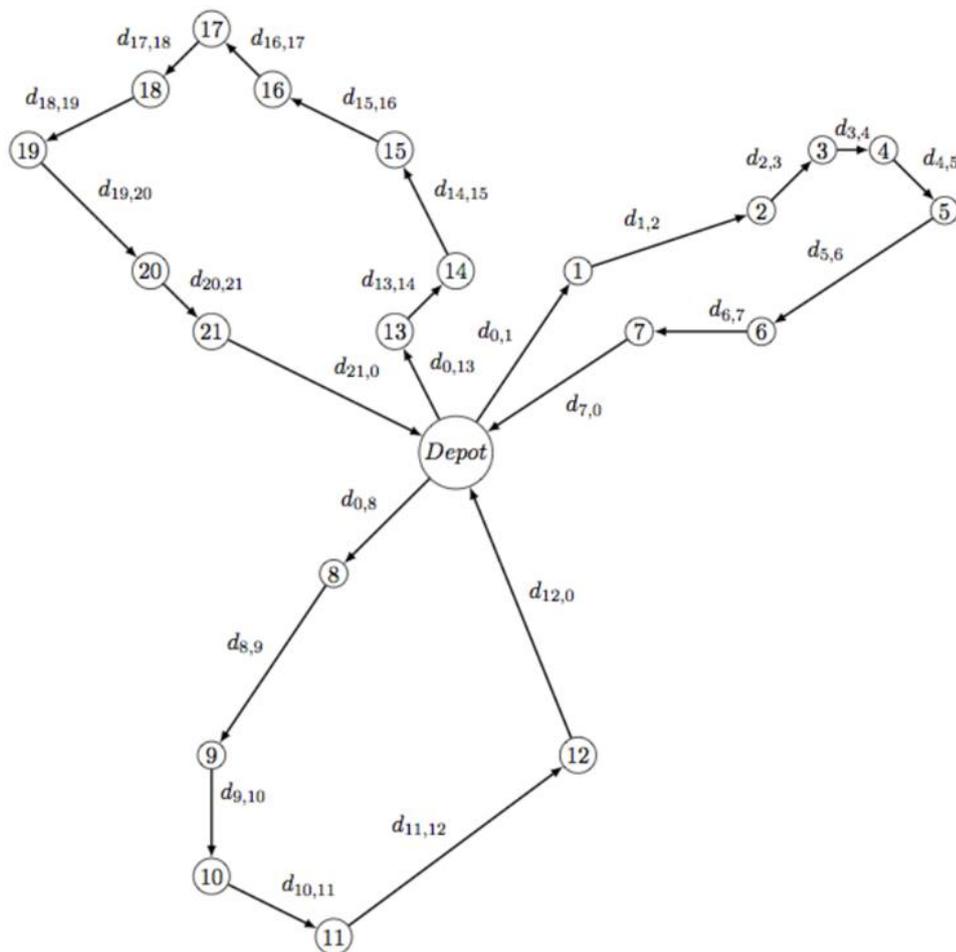
Formalmente, o CVRP assume que um conjunto Ω de $n + 1$ nós são numerados de 0 até n , sendo o nó 0 representado pelo depósito e os restantes dos nós as visitas a serem realizadas ($\Omega^* = \Omega - \{0\}$). Uma demanda $q_i > 0$ é atribuída a cada nó não-depósito i , sendo $i \in \Omega^*$, assumindo que $q_0 = 0$. Por outro lado, $A = \{(i,j) : i,j \in \Omega ; i < j\}$ representa o conjunto de $n \cdot (n + 1) / 2$ arestas existentes que conectam os $n + 1$ nós. Cada um desses links possuem um custo associado $C_{ij} > 0$ que representa o custo de enviar um veículo do nó i para o nó j . Nessa versão original do CVRP, os valores de C_{ij} são assumidos como simétricos ($C_{ij} = C_{ji}, 0 \leq i, j \leq n$), e são frequentemente expressos em termos da distância Euclidiana (d_{ij}) entre os dois nós. O processo de entrega deve ser realizado por uma frota de V veículos ($V \geq 1$) de igual capacidade, $Q \gg \max\{q_i : i \in \Omega\}$. Sendo os V veículos responsáveis pelas R rotas, ($R \leq V$) (CACERES-CRUZ et al., 2015). A Figura 2.1 apresenta um exemplo do CVRP.

Algumas restrições adicionais associadas à CVRP são apresentadas a seguintes (LAPORTE et al., 2000):

- Cada nó não-depósito é atendido por um único veículo.

- Todos os veículos começam e terminam suas rotas no depósito (nó 0).
- Um veículo não pode parar duas vezes no mesmo nó não-depósito.
- Nenhum veículo pode ser carregado excedendo a sua capacidade máxima.

Figura 2 - Representação de um exemplo do CVRP usando um depósito de distribuição central.



Fonte: Caceres-Cruz et al.(2015).

2.1.6 Open Vehicle Routing Problem (OVRP)

O OVRP é uma variante do problema clássico do roteamento do veículo (VRP), difere na medida em que os veículos não retornam ao depósito depois de atender os clientes. O OVRP surge em vários casos práticos, por exemplo, quando uma empresa não possui uma frota de veículos, ou sua frota privada é inadequada para satisfazer plenamente a demanda do cliente, a empresa deve alugar alguns veículos para satisfazer a demanda dos clientes. Uma vez que os veículos contratados não precisam retornar ao depósito da empresa contratante após a conclusão da tarefa, o roteamento é dado como aberto (LIU; JIANG, GENG, 2014). Além disso, o OVRP geralmente tem dois objetivos de otimização: minimizando o número de veículos usados e (dado este número de veículos) minimizando a distância total (ou às vezes o tempo) percorrida para o atendimento dos clientes (CACERES-CRUZ et al., 2015).

OVRP também segue as restrições básicas definidas para o CVRP, em que cada cliente é visitado uma única vez e a demanda total dos clientes atribuídos a uma rota não deve exceder a capacidade do veículo; e o custo geral de viagem do conjunto de rotas seja minimizado (LIU; JIANG, GENG, 2014).

2.1.7 VRP with Time Windows (VRPTW)

O problema de roteirização e programação de veículo com janela de tempo é uma generalização do problema clássico VRP adicionando-se as restrições de janela de tempo, em que as entregas a um determinado cliente devem ocorrer em um determinado intervalo de tempo, que varia de cliente para cliente (BRAEKERS; RAMAEKERS, VAN NIEUWENHUYSE, 2016).

No VRPTW cada cliente é associado a um intervalo de tempo e só pode ser atendido dentro desse intervalo. Neste problema, a dimensão do tempo é introduzida e, além disso, é preciso considerar o tempo de viagem e o tempo de atendimento em cada cliente. Como complemento, um conjunto de janelas de tempo também poderia ser considerado, variando esse modelo para um VRP de múltiplas janelas e tempo (*VRP with Multiple Time-Windows*) (VIDAL et al., 2013b).

As janelas de tempo também podem ser flexíveis dependendo de alguns custos extras (*VRP with Soft Time-Windows*). O problema de roteirização de veículos com janelas flexíveis é semelhante ao problema VRPTW, porém, neste caso, as janelas de tempo podem ser violadas mediante o pagamento de penalidades (VIDAL *et al.*, 2013b).

2.1.8 Pickup-and-Delivery VRP (PDVRP)

Nesta extensão do VRP, cada cliente está associado por duas quantidades, representando uma demanda a ser entregue no cliente e outra demanda para ser colhida e levada ao depósito. Como critério, as retiradas e entregas de mercadorias devem ser feitas pelo mesmo veículo, razão pela qual o local de recolhimento e o local de entrega devem ser incluídos na mesma rota. Adicionalmente, o total de recolhimento e o total de entregas em uma rota não podem exceder a capacidade do veículo e também deve garantir que essa capacidade não seja excedida em qualquer ponto da rota. Outra restrição é que a tarefa de coleta de um cliente deve preceder a de entrega (TASAN; GEN, 2012).

Uma variante do PDVRP é quando a demanda de retirada não é devolvida ao depósito, mas deve ser entregue a outro cliente, isto é, as cargas são coletadas em um nó e entregues em outro nó, resultando em uma relação de precedência entre as tarefas de recolher e entregar itens (LAPORTE, 2009).

2.1.9 Stochastic VRP (SVRP)

Existe um aspecto realista do problema de roteamento onde um comportamento aleatório é considerado, que geralmente é a presença de um cliente, sua demanda, seu tempo de serviço ou o tempo de viagem entre os clientes. As demandas dos

clientes podem mudar durante a aplicação de uma solução de roteamento, acarretando em uma aleatoriedade no problema.

No SVRP, duas etapas são realizadas para se obter uma solução. A primeira é determinada antes de conhecer as realizações das variáveis aleatórias. Numa segunda etapa, um recurso ou ação corretiva pode ser tomada quando os valores das variáveis aleatórias são conhecidos.

2.1.10 Green VRP (GVRP)

Essa abordagem introduz uma versão “verde” do tradicional VRP. Essa nova versão tem como objetivo incluir diferentes questões ambientais no processo de otimização, como, por exemplo, reduzir as emissões de gases de efeito estufa, poluição, desperdício e ruído durante o processo de realização da rota de entrega de mercadorias.

A função objetiva do problema de otimização avaliado é dada pela quantidade esperada de consumo de combustível. Aqui, o consumo de combustível é determinado com base em valores realistas para o consumo de combustível de caminhões existentes na frota, concentrando em dois fatores: o tipo de veículo utilizado e a quantidade de carga a ser carregada em cada veículo. Ambos os fatores são determinados pelos planejadores quando estão construindo os planos de rotas, implicando decisões sobre a sequência das rotas e sobre o uso dos veículos (KOPFER; SCHÖNBERGER, KOPFER, H., 2014).

2.2 PRINCIPAIS RESTRIÇÕES EXISTENTES EM VRPS

Os tipos de VRPs apresentados nas seções anteriores são classificados conforme suas restrições de roteamento. Para isso, Vidal et al. (2013b) propõem três grupos que representam os aspectos simples que qualquer solucionador deve lidar: atribuição de clientes e rotas por recursos, as escolhas da sequência e a avaliação das sequências fixas.

Para Lahyani, Khemakhem e Semet (2012), as restrições dos VRPs estão associadas aos níveis de decisão da empresa (operacional, tático e estratégico). O primeiro nível (estratégico) inclui decisões relacionadas aos locais, a número de depósitos utilizados e o tipo de dados. O nível tático define o tipo de ordem e as frequências de visitas dos clientes em um determinado horizonte de tempo. Finalmente, o operacional considera os horários dos veículos e do motorista, de modo que as restrições estão relacionadas ao planejamento da distribuição e especificadas para clientes, veículos, motoristas e estradas. Além disso, os autores propõem um segundo nível de classificação associado ao elemento de roteamento envolvido (depósito, cliente, rota, veículo e produto) para ajudar uma melhor compreensão da classificação.

As listagens a seguir apresenta as principais restrições de distribuição existentes em problemas de roteamento consideradas nos trabalhos (CACERES-CRUZ et al., 2015), (BRAEKERS et al., 2016), (VIDAL, 2013b), (LAHYANI; KHEMAKHEM; SEMET, 2012).

Restrições da carga:

- Multiprodutos (*Multi-Products – CP*): Alguns veículos podem transportar vários tipos de produtos (fresco, congelado, pequeno, grande, etc.).
- Capacidade Multidimensional (*Multi-Dimensional capacity – CD*): A capacidade dos veículos é considerada em 2D ou 3D.
- Capacidade do Veículo (*Vehicle Capacity – C*): A capacidade dos veículos é limitada.

Restrições da frota:

- Frota Homogênea de Veículos (*Homogeneous Fleet of Vehicles – FO*): Todos os veículos da frota possuem a mesma capacidade.
- Frota Heterogênea de Veículos (*Heterogeneous Fleet of Vehicles - FE*): Vários tipos de veículos (capacidades) podem ser encontrados na frota.
- Frota de Veículos Não Fixada (*Unfixed Fleet of Vehicles – VU*): O número de veículos existe na frota é considerado é ilimitado.
- Frota Fixa de Veículos (*Fixed Fleet of Vehicles – VF*): O número de veículos existe na frota considerado é limitado.
- Custo Fixo por Veículo (*Fixed Cost per Vehicle – FC*): usar um veículo implica um custo extra.
- Custo Variável de Veículo (*Variable Cost of Vehicle – VC*): O custo real é representado pelo produto da distância atribuída a um veículo e seu preço por unidade de distância.
- Múltiplas Viagens (*Multi-Trips – MT*): Todos ou alguns veículos da frota podem executar mais de uma viagem (múltiplas utilizações de veículos).
- Reabastecimento durante a rota (*Intra-route Replenishments – IR*): os veículos devem ser recarregados em alguns pontos durante a realizações dos percursos.

Restrições do ambiente:

- Veículo Dependente da Localidade (*Vehicle Site Dependent – DS*): alguns veículos não podem visitar alguns pontos devido a problemas geográficos, de compatibilidade ou legais.
- Veículos Dependente da Estrada (*Vehicle Road Dependent - DR*): Alguns veículos não podem passar por algumas ruas, avenidas, rodovias e estradas

devido aos seus limites e problemas legais relacionados ao seu tamanho, carga e capacidade.

- Restrições de Duração/Comprimento (*Duration Constraints/Length – L*): a duração de cada rota não pode exceder um valor ou custo máximo, incluindo os tempos de serviço em cada cliente visitado.

Restrições trabalhistas:

- Mudanças de motorista e regulamento de trabalho (*Driver Shifts/Working Regulations – D*): O planejamento das rotas inclui o número de horas de trabalho legais dos motoristas, além das paradas, quebras, descanso, etc.
- Rotas Equilibradas (*Balanced Routes – BR*): A rotas deve ser equilibrada entre todos os veículos. O balanceamento pode ser definido pelo número de clientes a serem visitados, pelas distâncias das rotas e/ou por suas durações estimadas.

Restrições do ambiente:

- Matriz de Custo Simétrico (*Symmetric Cost Matrix – CS*): A matriz de custos tem uma natureza simétrica.
- Matriz de Custo Assimétrico (*Asymmetric Cost Matrix – CA*): A matriz de custo tem uma natureza assimétrica.

Restrições do cliente:

- Tempo Dependente, Dinâmico e Estocástico (*Time Dependent/Dynamic/Stochastic times - TD*): O objetivo é minimizar o tempo de viagem, sendo que ele pode variar ao longo do dia (fixo ou flexível). Além disso, a localização/distância dos clientes também pode muda.
- Demandas Estocásticas/Dinâmicas (*Stochastic Demands/Dynamic – S*): as demandas dos clientes podem mudar durante a aplicação de uma solução de roteamento.

- Janela de Tempo (*Time Windows – TW*): os clientes não podem receber as suas encomendas fora de determinado momento. Cada cliente possui uma janela de tempo particular (fixo ou flexível).
- Múltiplas Janelas de Tempo (*Multiple Time Windows – MW*): os clientes não podem receber os pedidos fora de um conjunto de intervalos de tempo. Cada cliente possui um determinado conjunto de janelas de tempo.

Restrições cliente/serviço:

- Coleta e Entregar (*Pick-up & Delivery - PD*): A construção de rotas deve considerar as retiradas de produtos em alguns clientes e a entrega a outros, de forma sequencial ou separada. O depósito apenas define o ponto inicial e final dos veículos.
- Coleta e Entregar Simultânea (*Simultaneous Pick-up & Delivery – PS*): A construção das rotas deve considerar o recolhimento e entrega de produtos/pessoas ao mesmo tempo para todos os nós de um mesmo veículo. O depósito apenas define o ponto inicial e final dos veículos.
- *Backhauls (B)*: A construção de rotas deve considerar a retirada de produtos em alguns clientes e a entrega a outros, de forma sequencial ou separada. O pressuposto crítico é que todas as entregas devem ser feitas em cada rota antes de qualquer retirada pode ser realizada (às vezes, um cliente pode exigir tanto uma entrega quanto uma coleta). O rearranjo de produtos pode ser caro ou inviável. O depósito apenas define o ponto inicial e final dos veículos.
- Visitas Múltiplas/Entregas divididas (*Multiple Visits/Split deliveries – MV*): Os clientes são visitados várias vezes por entregar o resumo das ordens originais. Cada veículo pode fornecer uma fração da demanda do cliente.
- Múltiplos Período/Periódico (*Multi-Period/Periodic – MP*): A otimização é feita ao longo de um conjunto de dias, considerando várias visitas. Além disso, que cada cliente pode ter uma frequência de visitas diferente.

- Controles dos níveis de inventário (*Inventory Levels Controls – I*): Os custos dos armazenamentos também são considerados minimizados com os custos de roteamento enquanto os níveis de estoque são controlados.
- Capacidade do Cliente (*Customer Capacity – CC*): Quando o estoque de capacidade dos clientes também é considerado.
- Restrições de Precedência (*Precedence constraints – PC*): A ordem de visitas dos clientes pode ser importante para o carregamento e descarregamento de produtos. Seu pedido pode ser importante por motivos saudáveis ou de segurança.

Restrições da empresa/regras de negócio:

- Múltiplos Depósitos (*Multi-Depot MD*): Existem mais de um depósito de onde os veículos saem e chegam.
- Janela de Tempo do Depósito (*Time Windows for the Depot - WD*): O depósito fica aberto durante um período de tempo. Então, se os veículos precisam fazer mais de uma viagem é necessário considerar o intervalo em que o depósito permanece aberto.
- Diferentes Locais de Destino/Rota Aberta (*Different end locations/Open Routes – O*): As rotas começam no depósito, mas terminam no último cliente. O custo de retorno não é considerado e pode ser opcional.
- Diferentes Locais de Início e Término (*Different start and end locations – DA*): Os veículos começam e terminam em diferentes locais.
- Partida de Diferentes Locais (*Departure from Different Locations – DD*): Os veículos começam em locais diferentes.
- Multi-objetivos (*Multi-Objectives – MO*): O planejamento da rota considera mais de uma função objetiva ou custos relacionados ao mesmo tempo.

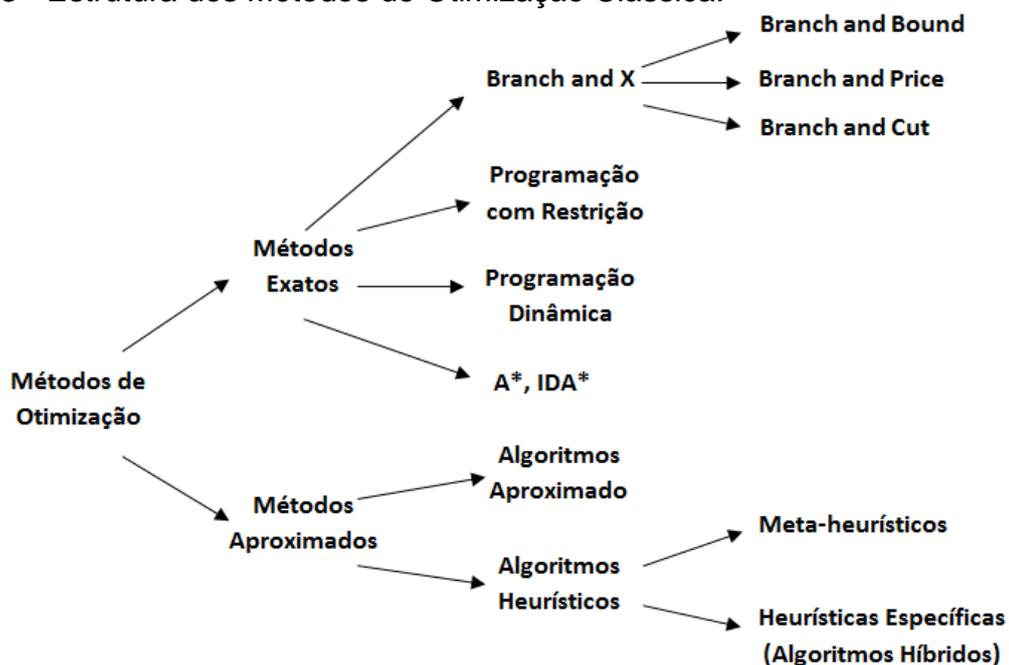
2.3 METODOLOGIAS DE RESOLUÇÃO PARA O VRP

Diferentes abordagens para a resolução de problemas de roteamento veiculares têm sido exploradas, que variam desde o uso de métodos exatos à métodos heurísticos e híbridos.

Enquanto os métodos exatos, como a programação matemática, tentam resolver problemas de pequeno a médio porte (cerca de 75 a 100 clientes) contendo restrições relativamente simples, os métodos heurísticos e híbridos oferecem soluções de boa qualidade para problemas de tamanho médio e grande, contendo restrições mais complexas.

As heurísticas atendem a três objetivos principais: resolver problemas mais rapidamente, resolver problemas maiores e permitem a elaboração de algoritmos mais robustos, provendo soluções aceitáveis em um tempo razoável na resolução de problemas difíceis e complexo. Seguindo a divisão proposta por (TALBI, 2009), uma grande família de métodos de otimização pode ser resumida na Figura 3.

Figura 3 - Estrutura dos Métodos de Otimização Clássica.



Fonte: Próprio autor. Adaptado de Caceres-Cruz et al. (2015).

2.3.1 Métodos Exatos

Os métodos exatos obtêm soluções ótimas. Esse tipo de técnica é frequentemente aplicado a instâncias de pequeno porte, pois em dimensões elevadas, torna-se impraticável uma enumeração completa de todas as possíveis soluções do problema, demandando um tempo muito elevado de processamento (TALBI, 2009).

A família dos métodos exatos de otimização possui um amplo conjunto de diferentes métodos que podem ser aplicados na resolução do problema de VRP, como o proposto por Christofides em Balas (1981) e Christofides, Mingozzi e Toth (1981), que apresentou pela primeira vez a implementação de algoritmos exatos baseados em Relaxação Lagrangeana na resolução de problemas de otimização combinatória, tendo como essência a resolução de subproblemas com um modelo independente.

2.3.2 Métodos aproximados ou heurísticos

Métodos aproximados são algoritmos que não necessariamente produzem uma solução ótima, mas soluções que estão dentro de um certo fator da solução ótima, tendo como objetivo diminuir ao máximo o intervalo (*gap*) existente entre a solução ótima e os resultados encontrados.

O desenvolvimento de algoritmos de aproximação surgiu devido à impossibilidade de se resolver otimamente diversos problemas de otimização NP-difícil, que se refere à impossibilidade de encontrar algoritmos eficientes para a resolução de problemas de buscas.

Pelo fato do VRP ser um problema NP-difícil, é razoável sacrificar a otimalidade em troca da garantia de uma solução aproximada que seja computacionalmente resolvível. Sendo que, apesar de sacrificar o resultado ótimo, a solução tenha boas

garantias sobre os resultados obtidos, visando ganhar o máximo em termos de eficiência computacional (MALAQUIAS et al., 2016).

Sendo esse compromisso, entre a perda de otimalidade e ganho de eficiência, o paradigma dos algoritmos de aproximação.

2.3.3 Métodos heurísticos e meta-heurísticos

Heurísticas são processos cognitivos empregados em decisões não racionais, sendo definidas como estratégias que ignoram parte da informação com o objetivo de tornar a escolha mais fácil e rápida (GIGERENZER; GAISSMAIER, 2011). Os métodos heurísticos são denominações para os algoritmos que fornecem soluções sem um limite formal de qualidade, tipicamente avaliado empiricamente em termos de complexidade e qualidade das soluções.

As heurísticas encontram boas soluções em instâncias de problemas de grande porte. Elas permitem obter um desempenho com custos aceitáveis em uma ampla gama de problemas, apesar de não possuírem uma aproximação garantida nas soluções obtidas. Elas são adaptadas e projetadas para resolver problemas específicos (TALBI, 2009).

Tendo como ponto de partida uma solução, esses métodos baseiam-se em sucessivas aproximações direcionadas a um ponto que seja factível. Logo, costumam encontrar boas soluções que sejam factíveis, com um tempo computacional viável. Assim são os métodos heurísticos: uma busca contínua e empírica, com vários ótimos locais, cujo resultado é o melhor que se pode encontrar sob determinadas condições.

Atualmente, a maioria dos métodos desenvolvidos para o VRP, encontrados na literatura, é heurística, devido ao tempo de resolução e sua capacidade de lidar com grandes instâncias. A tradição determina que uma função é usada quando as heurísticas são aplicadas: a primeira prioridade é minimizar o número de veículos

usados e o segundo para minimizar o custo das viagens de cada veículo (TOTH; VIGO, 2014).

As meta-heurísticas são algoritmos de propósito geral que podem ser aplicados para resolver vários tipos de problemas de otimização. Eles podem ser vistos como metodologias gerais de nível superior que podem ser usadas como uma estratégia orientadora na concepção de heurísticas subjacentes. Talbi (2009) também propõe que dois critérios contraditórios sejam levados em consideração: exploração do espaço de busca (diversificação) e exploração das melhores soluções encontradas (intensificação).

As regiões promissoras são determinadas pelas boas soluções obtidas. Na intensificação, as regiões promissoras são exploradas com maior intensidade na esperança de encontrar melhores soluções. Enquanto que, na diversificação, as regiões não exploradas devem ser visitadas para ter certeza de que todas as regiões do espaço de busca são verificadas uniformemente e que a busca não se limita a apenas um número reduzido de regiões.

Existem muitas meta-heurísticas inspiradas em processos naturais como Algoritmos Evolutivos (incluindo Algoritmos Genéticos) e Otimização de Colônias de Formigas (ACO). Esses algoritmos são baseados em populações e indivíduos que a cada iteração sofrem um melhoramento ou adaptação com o objetivo de encontrar melhores soluções e se aproximar da ideal.

2.4 PARÂMETROS DOS MÉTODOS DE OTIMIZAÇÃO

Os problemas de otimização são problemas de maximização ou minimização da função de uma ou mais variáveis de um determinado domínio, sendo que, geralmente, existe um conjunto de restrições nas variáveis.

Para melhor entendimento dos algoritmos de otimização, faz-se necessário o conhecimento de alguns conceitos e definições utilizados na literatura. A seguir são

listados alguns termos usualmente relacionados a um problema de otimização qualquer:

- Variáveis de decisão: são aquelas que se alteram durante o processo de otimização, podendo ser contínuas (reais), inteiras ou binárias.
- Restrições: são funções de igualdade ou desigualdade sobre as variáveis de projeto que descrevem situações de projeto consideradas não desejáveis.
- Espaço de busca: é o conjunto, espaço ou região que compreende as soluções possíveis ou viáveis sobre as variáveis do projeto do problema a ser otimizado, sendo delimitado pelas funções de restrição.
- Função Objetivo: é a função de uma ou mais variáveis de projeto que se quer otimizar, minimizando-a ou maximizando-a.
- Ponto Ótimo: é o ponto formado pelas variáveis de projeto que extremizam a função objetivo e satisfazem as restrições.
- Valor Ótimo: é o valor da função objetivo no ponto ótimo.

A seguir são apresentadas as possíveis métricas utilizadas para o problema de roteamento de veículos.

Função Objetivo

- Minimizar a distância total percorrida pela frota de veículos;
- Minimizar o número de veículos utilizados no roteamento;
- Minimizar os custos totais de distribuição, que incluem os custos fixos (custo de capital do veículo, salário de motoristas e ajudantes, despesas com seguros, taxas, impostos etc.).
- Maximizar a função utilidade baseada no nível de serviço e/ou prioridades dos clientes.

As principais restrições são descritas a seguir:

Restrições dos Veículos

- Limite de capacidade dos veículos;
- Limite com relação ao tipo de carga dos veículos;
- Operação de carga e descarga dos veículos;
- Números veículos disponíveis;
- Tipos de veículos disponíveis.

Restrições das rotas

- Horário de início e término das viagens;
- Tempo máximo de viagem de um veículo;
- Distância máxima percorrida;
- Locais de paradas fixas.

Restrições com os clientes

- Janela de tempo dos clientes;
- Atendimento total ou parcial das demandas;
- Tempo máximo permitido para carga e descarga;
- Necessidade ou restrição de serviço em algum dia específico da semana;
- Disponibilidade de área para estacionamento do veículo.

As variáveis de decisão são detalhadas a seguir:

Variáveis de decisão

- Roteiro a ser percorrido por cada veículo;
- Qual veículo é designado para cada cliente;
- Qual a quantidade de carga transportada para cada cliente da rota;
- Tempo de início do atendimento do primeiro cliente da rota;
- Tempo de finalização do atendimento do primeiro cliente da rota.

A seguir estão detalhadas as principais características do problema:

Tipo de operação

- Entrega;
- Coleta;
- Coleta e entrega simultaneamente;
- Coleta (ou entrega) com carga de retorno;
- Realização de algum tipo de serviço no cliente.

Tamanho da frota

- Limitada;
- Ilimitada.

Tipo de frota

- Homogênea;
- Heterogênea.

Depósito e localização de veículos

- Um único depósito;
- Vários depósitos;
- Quantidade de produtos disponíveis no depósito central para entrega aos clientes;
- Número de bases de origem e destino dos veículos.

Tipo de carga

- Única ou carga de lotação;
- Múltiplas cargas ou carga fracionada.

Tipo de demanda

- Determinística;
- Estocástica.

Localização da demanda

- Demanda localizada somente em arcos;
- Demanda localizada somente em nós;
- Demanda localizada em arcos e nós.

Jornada de trabalho

- Duração;
- Horário de almoço e outras interrupções;
- Permissão para viagem com mais de um dia de duração;

- Número de tripulação por veículo.

Pagamento dos tripulantes

- Por jornada de trabalho;
- Por produtividade;
- Jornada e horas extras.

Visita e atendimento aos clientes

- Cada veículo visita um cliente uma única vez durante a rota;
- Um cliente pertence a uma única rota;
- Um cliente pode pertencer a mais de uma rota;
- Um cliente pode ser atendido mais de uma vez durante a rota.

Estrutura da rede

- Direcionada;
- Não direcionada;
- Mista;
- Euclidiana.

Horizonte de planejamento

- Curto prazo;
- Longo prazo.

3 ALGORITMOS GENÉTICOS E *SPACE-FILLING CURVES*

Este capítulo apresenta a fundamentação teórica necessária para o entendimento desse trabalho, incluindo os algoritmos e os problemas conhecidos na literatura, bem como as tecnologias que foram utilizadas na implementação das soluções propostas.

3.1 ALGORITMOS GENÉTICOS

Os Algoritmos Genéticos (AG) (HOLLAND, 1992) pertencem a uma categoria de algoritmos computacionais evolucionários inclusos nas técnicas de Inteligência artificial (IA), os quais são fundamentados no processo de seleção natural que buscam simular a evolução biológica natural dos seres vivos, se adaptando a cada geração ao meio ambiente, garantido, dessa forma, maior chance de sobreviver e gerar descendentes (LINDEN, 2012). Esse processo natural básico foi originalmente descrito por Darwin com base em três princípios básicos: reprodução, seleção natural e diversidade de indivíduos.

Cada indivíduo na população representa uma solução particular para o nosso problema e esse indivíduo é representado com um conjunto de propriedades que estão escritas no cromossomo ou no genótipo do indivíduo, bem como na natureza. Os processos de evolução e seleção natural são computados sobre a população de soluções candidatas (indivíduos). Com base em seu genótipo, cada indivíduo é avaliado com o que é chamado de uma função de adaptação (*fitness function*), e somente os indivíduos mais adaptados conseguem prosperar (KARAKATIČ; PODGORELEC, 2015).

Algoritmos Genéticos são muito eficientes para busca de soluções aproximadamente ótimas em uma grande variedade de problemas, uma vez que não impõem muitas das limitações encontradas nos métodos de busca tradicionais, pois operam sobre

uma população de candidatos em paralelo, realizando buscas em diferentes áreas do espaço de solução, alocando um número de indivíduos apropriado para a busca em diferentes regiões (GOLDBERG, 1989). De outra maneira, esse algoritmo difere dos métodos tradicionais de busca e otimização, principalmente em quatro aspectos (RAIMANN, 2007):

- AGs trabalham com uma codificação do conjunto de parâmetros e não com os próprios parâmetros;
- AGs trabalham com uma população e não com um único ponto;
- AGs utilizam informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar;
- AGs utilizam regras de transição probabilísticas (na seleção aleatória) e não determinísticas (na seleção individual).

Na implementação do algoritmo, inicialmente é gerada uma população formada por um conjunto aleatório de indivíduos que podem ser vistos como possíveis soluções do problema. Durante o processo evolutivo, esta população é avaliada, ou seja, para cada indivíduo é dada uma nota, ou índice, refletindo sua habilidade de adaptação a determinado ambiente (LINDEN, 2012). Uma porcentagem dos mais adaptados é mantida, enquanto os outros são descartados (darwinismo).

Os membros mantidos pela seleção podem sofrer modificações em suas características fundamentais através de mutações e cruzamento (*crossover*) ou recombinação genética gerando descendentes para a próxima geração. Este processo, chamado de reprodução, é repetido até que uma solução satisfatória seja encontrada (MITCHELL, 1998). Nas próximas seções serão apresentados os conceitos do algoritmo detalhadamente.

3.1.1 Características gerais dos algoritmos genéticos

Os GAs são algoritmos de otimização global, baseados nos mecanismos de seleção natural e da genética. Eles empregam uma estratégia de busca paralela e estruturada, mas aleatória, que é voltada em direção ao reforço da busca de pontos de "alta aptidão", ou seja, pontos nos quais a função a ser minimizada (ou maximizada) tem valores relativamente baixos (ou altos) (KARAKATIČ; PODGORELEC, 2015).

Apesar de sua natureza estocástica, eles não são considerados caminhos aleatórios não direcionados, pois exploram informações históricas para encontrar novos pontos de busca onde são esperados melhores desempenhos. Isto é feito através de processos iterativos, onde cada iteração é chamada de geração (DENG; LIU; ZHOU, 2015).

Durante cada iteração, os princípios de seleção e reprodução são aplicados a uma população de candidatos que pode variar, dependendo da complexidade do problema e dos recursos computacionais disponíveis. Através da seleção, é possível verificar quais indivíduos conseguirão se reproduzir, gerando certo número de descendentes para a próxima geração, com uma probabilidade estipulada pelo seu índice de aptidão. Os indivíduos com maior adaptação relativa terão maiores chances de se reproduzir (LINDEN, 2012).

Tradicionalmente, os indivíduos são representados por vetores binários, no qual cada elemento de um vetor denota a presença ou ausência de uma determinada característica: o seu genótipo. Os elementos podem ser combinados formando as características reais do indivíduo, ou o seu fenótipo. Teoricamente, esta representação é independente do problema, pois uma vez encontrada a representação em vetores binários, as operações padrão podem ser utilizadas, facilitando o seu emprego em diferentes classes de problemas (MITCHELL, 1998).

Especificamente em problemas de rotas, os indivíduos são formados por genótipos que representam uma série de paradas. Essas paradas são pontos (clientes) que serão visitados um após o outro, conforme a sequência determinada no vetor de características do indivíduo.

Dentro do AG existem algumas definições básicas a serem consideradas (KARAKATIČ, 2015) (RAIMANN, 2007):

- **Indivíduos:** simples membro da população. Nos Algoritmos Genéticos, um indivíduo é formado pelo cromossomo e sua aptidão, e representa cada candidato à solução do problema. O processo de solução adotado pelos Algoritmos Genéticos consiste em gerar, por meio de regras específicas, um grande número de indivíduos de forma a promover uma varredura tão extensa quanto necessária do espaço de busca do problema.
- **População:** conjunto de indivíduos ou soluções.
- **Gene:** elemento que compõe os cromossomos. Nos Algoritmos Genéticos, é unidade básica do cromossomo representada por um símbolo. Cada cromossomo tem certa quantidade de genes, descrevendo uma variável de problema.
- **Cromossomo:** elemento portador do material genético. Nos Algoritmos Genéticos representam uma cadeia de bits associada a uma solução possível para o problema. Deve ser observada que cada cromossomo corresponde a um ponto no espaço de solução do problema de otimização.
- **Genótipo:** conjunto de genes de um indivíduo.
- **Fenótipo:** conjunto de características de um indivíduo determinadas pelo genótipo.
- **Locus (posição):** posição em que um gene se encontra no cromossomo.
- **Alelo:** diferentes possibilidades de uma característica de um gene. Nos Algoritmos Genéticos representa o valor que um gene pode assumir, ou seja, é cada símbolo utilizado para codificação.
- **Geração:** é cada passo do processo evolutivo. Considerada o ciclo de criação e transformação de uma população, representada pelo número da iteração que os Algoritmos Genéticos executam.

- **Operações Genéticas:** operações que os Algoritmos Genéticos realizam sobre as estruturas dos cromossomos. Tais operações visam promover a evolução do indivíduo.
- **Espaço de Busca:** é o conjunto espaço ou região que compreende as soluções possíveis ou viáveis para o problema a ser resolvido.
- **Aptidão ou Adequação (*Fitness*):** representa a informação numérica do desempenho de cada indivíduo da população e está associada à função objetivo e às restrições do problema. Por meio da adequação ou aptidão é possível selecionar os melhores indivíduos de cada população para aplicação das operações genéticas.

É importante também, analisar de que maneira alguns parâmetros influenciam no comportamento dos Algoritmos Genéticos, para que se possa estabelecê-los conforme as necessidades do problema e dos recursos disponíveis. São listados a seguir alguns Parâmetros Genéticos utilizados:

- **Tamanho da População:** O tamanho da população afeta o desempenho global e a eficiência dos AGs. Com uma população pequena o desempenho pode cair, pois deste modo a população fornece uma pequena cobertura do espaço de busca do problema. Uma grande população geralmente fornece uma cobertura mais representativa do domínio do problema, além de prevenir convergências prematuras para soluções locais ao invés de globais. No entanto, para se trabalhar com grandes populações, são necessários maiores recursos computacionais, ou que o algoritmo trabalhe por um período de tempo muito maior.
- **Tipo de Cruzamento:** O tipo de cruzamento a ser utilizado determina a forma como se procederá a troca de segmentos de informações entre os “casais” de indivíduos selecionados, impactando, conseqüentemente, na formação dos indivíduos das próximas gerações.
- **Taxa de Cruzamento:** Quanto maior for esta taxa, mais rapidamente novas estruturas serão introduzidas na população. Mas se esta for muito alta, a maior

parte da população será substituída, podendo ocorrer perda de estruturas de alta aptidão. Com um valor baixo, o algoritmo pode tornar-se muito lento.

- **Taxa de Mutação:** Mutação é utilizada para dar nova informação para a população e também para prevenir que ela se sature como cromossomos semelhantes. Uma baixa taxa de mutação previne que uma dada posição fique estagnada em um valor, além de possibilitar que se chegue em qualquer ponto do espaço de busca. Com uma taxa muito alta a busca se torna essencialmente aleatória, além de aumentar muito a possibilidade de que uma boa solução seja destruída.

3.1.2 Métodos e critérios para a implementação do algoritmo genético

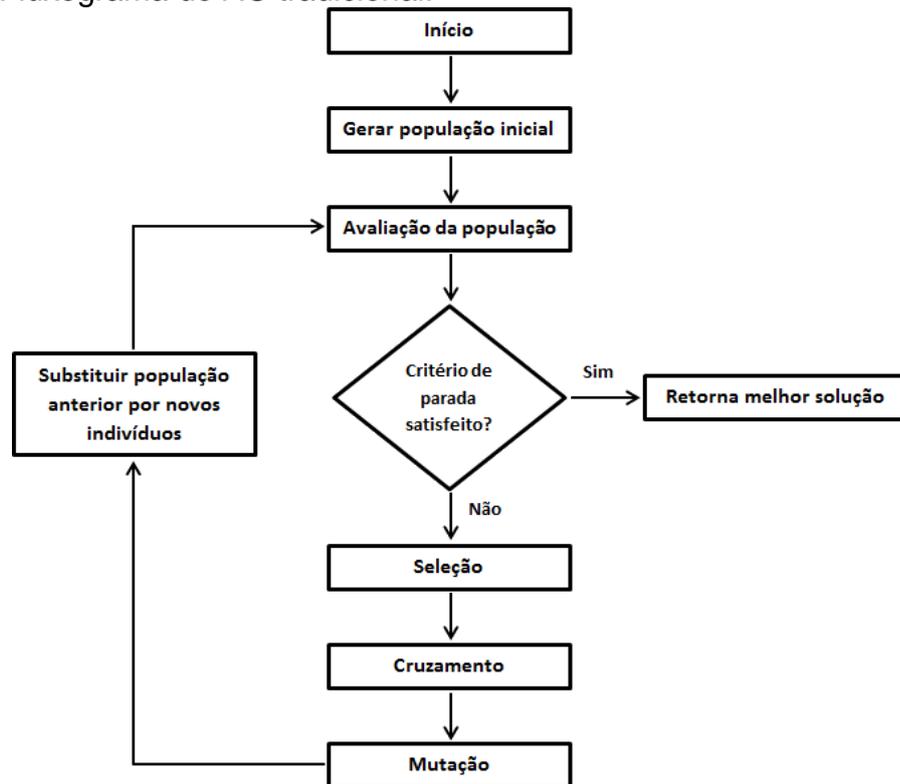
Para a implementação de um AG, é necessário definir de forma correta alguns métodos e critérios. A seleção de um método ou critério depende especificamente do tipo de problema a ser resolvido e também de que certos requisitos imprescindíveis não sejam violados. A seguir são listados alguns desses métodos e critérios (MITCHELL, 1998) (MCPHEE; POLI, LANGDON, 2008):

- **Critério de implementação:** tendo em vista que o AG trabalha com vetores de indivíduos, deve-se especificar a codificação com a qual se faz corresponder cada ponto do domínio do problema com um gene ou conjunto de genes do cromossomo.
- **Critério de tratamento dos indivíduos:** nem sempre é possível estabelecer uma correspondência ponto-a-ponto entre o domínio do problema e o conjunto de indivíduos criados para resolver tal problema. Como consequência, indivíduos menos adaptados não devem contribuir para a geração de populações futuras.
- **Critério de seleção:** a seleção deve dirigir o processo de busca em favor dos indivíduos mais aptos.

- **Critério de substituição:** os critérios com que se selecionam os indivíduos pais não, necessariamente, precisam ser os mesmos usados para selecionar os sobreviventes de cada geração.
- **Critério de inicialização:** refere-se a como deve ser construída a população inicial com a qual se iniciará o Algoritmo Genético.
- **Critério de parada:** devem ser determinadas as condições nas quais se considera que o Algoritmo Genético encontrou uma solução aceitável no processo de busca na resolução do problema e não faça sentido a continuação da execução do algoritmo.
- **Função de avaliação e aptidão:** deve ser determinada a função de avaliação mais apropriada na resolução o problema em questão, assim como a função de aptidão que utilizará o Algoritmo Genético para resolvê-lo.
- **Operadores Genéticos:** o Algoritmo Genético faz uso, normalmente, de três operadores genéticos, que são: seleção, cruzamento e mutação. Ficando claro que esses não são os únicos possíveis e ainda possibilitam variações.

O fluxograma a seguir mostra a estrutura básica da implementação do Algoritmo Genético tradicional.

Figura 4 - Fluxograma do AG tradicional.



Fonte: Próprio autor.

O pseudocódigo do Algoritmo Genético tradicional é mostrado na Tabela 3.

Tabela 3 - Pseudocódigo do Algoritmo de Genético.

```

Gerar população inicial P
i = 0
Enquanto condição de parada não satisfeita faça
  Avalie cada indivíduo da população
  Crie espaços para nova população NP = 0

  Enquanto |NP| <> |P| faça
    Selecionar pais a partir de P
    Gerar os indivíduos filhos a partir do cruzamento dos pais
    Aplicar mutação nos filhos
    Inserir filhos em NP
  Fim Enquanto
  P <-- NP
  i = i + 1
Fim enquanto
Retorne melhor indivíduo de P como solução
  
```

Fonte: Próprio autor.

3.1.3 Seleção de indivíduos

A operação de seleção é utilizada para selecionar os indivíduos que estão mais aptos para uma futura reprodução de seus descendentes, e que conseqüentemente apresentarão melhores soluções na resolução do problema. Para essa escolha utiliza-se o valor da aptidão (*fitness*) de cada indivíduo para o processo de reprodução (MCPHEE; POLI, LANGDON, 2008).

Após a seleção dos indivíduos mais aptos, a próxima etapa diz respeito como os cromossomos de cada indivíduo serão selecionados para sofrerem as operações de cruzamento e, conseqüentemente, gerar os descendentes que poderão sofrer o processo de mutação.

Esse processo desempenha o papel da seleção natural na evolução, selecionando para sobreviver e reproduzir os organismos melhores adaptados ao meio, no caso, os indivíduos com melhor valor na função de adequação.

Na operação de seleção, cada dois indivíduos selecionados, são gerados dois novos filhos por meio do processo de cruzamento. O processo de seleção termina quando o número de indivíduos se iguala ao da população anterior. Logo, um novo processo é desencadeado e ocorre a troca da população antiga pela nova, contendo indivíduos mais evoluídos e mais aptos na resolução do problema.

A maneira pela qual os indivíduos são selecionados para reprodução pode variar, dependendo do método de seleção utilizado. A seguir são descritos alguns métodos de seleção mais conhecidos (LINDEN, 2012) (MITCHELL, 1998) (MCPHEE; POLI, LANGDON, 2008).

3.1.3.1 Seleção aleatória

É o método mais simples: cada indivíduo tem a mesma probabilidade de ser escolhido.

$$\varphi_s = (x_i(t)) = \frac{1}{n}, \quad (2.1)$$

onde,

- φ_s é a probabilidade de cada indivíduo ser selecionado.
- n é o tamanho da população.
- x_i é o indivíduo i da iteração t .

No entanto, nesse método de seleção nenhuma informação sobre a avaliação de cada indivíduo é usada. Além disso, é o método com menor precisão seletiva.

3.1.3.2 Seleção por elitismo

Neste método os n melhores de cada geração são levados para a próxima geração.

Por exemplo, em uma geração de p indivíduos somente $p - n$ indivíduos precisam ser gerados por cruzamentos.

O ponto negativo desse método é que quanto mais indivíduos sobrevivem para a próxima geração menor será a diversidade genética da próxima geração.

3.1.3.3 Seleção por roleta

No método de Roleta, cada indivíduo tem a oportunidade de ser selecionado de acordo com seu desempenho relativo ao da população, ou seja, cada indivíduo da população é representado em uma roleta imaginária proporcionalmente ao seu índice de aptidão. Nesse processo, somam-se todos os valores das aptidões dos indivíduos de uma população. Para cada indivíduo divide-se suas aptidões pelo valor da soma total das aptidões de todos os indivíduos, chegando à probabilidade que cada um tem de ser selecionado dentro da população. A sequência de equações de 3.2 a 3.5 apresentam as etapas dos cálculos de probabilidade existente no método de Seleção por Roleta. A Equação 3.2 calcula a avaliação de cada indivíduo. Enquanto que a Equação 3.3 calcula a avaliação total da população. A Equação 3.4 calcula a probabilidade de seleção de cada indivíduo e a Equação 3.5 calcula a probabilidade cumulativa para cada indivíduo.

$$f_Y(x_i(t)) \quad (2.2)$$

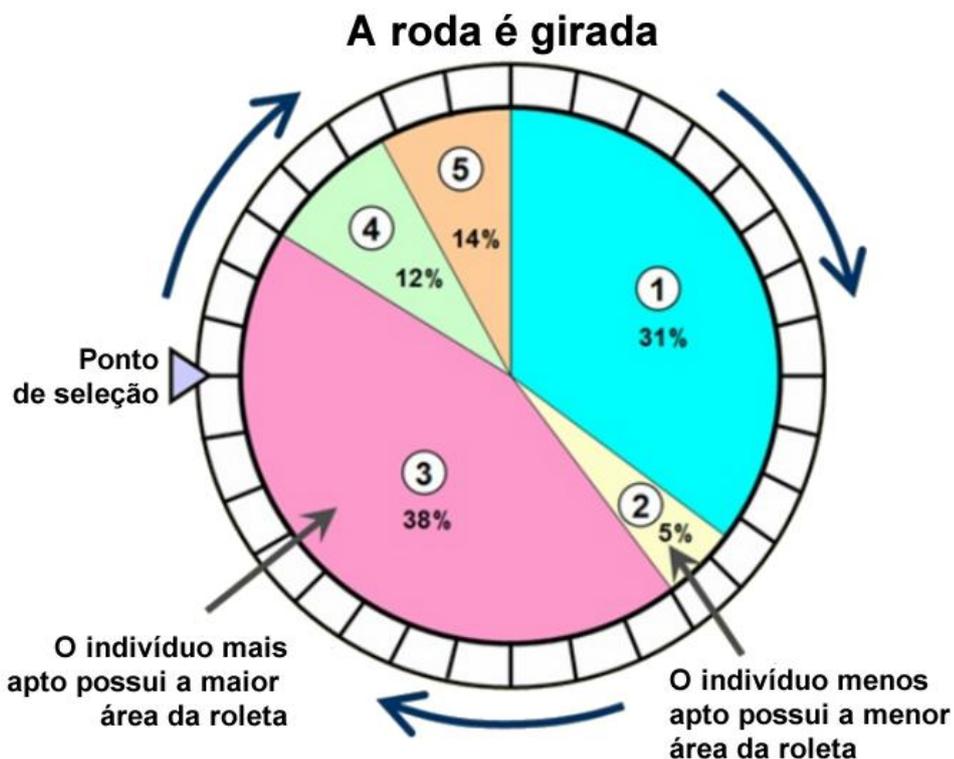
$$S = \sum_{l=0}^{n-1} f_Y(x_l(t)) \quad (2.3)$$

$$\varphi_s(x_i(t)) = \frac{f_Y(x_i(t))}{\sum_{l=0}^{n-1} f_Y(x_l(t))} \quad (2.4)$$

$$\theta_S(x_i(t)) = \sum_{l=0}^{n-1} \varphi_s(x_l(t)) \quad (2.5)$$

A roleta é construída com setores correspondentes aos valores de aptidão (*fitness*) de cada indivíduo. O tamanho de cada seletor correspondente é determinado de acordo com a probabilidade de seleção de cada indivíduo. Aos indivíduos com alta aptidão é dada uma proporção maior da roleta, enquanto aos de baixa aptidão é dada uma proporção relativamente menor. Assim, aqueles mais aptos terão maiores chances de serem sorteados. Finalmente, a roleta é girada um determinado número de vezes, dependendo do tamanho da população, e são escolhidos, como indivíduos que ganharão o direito de filhos, aqueles sorteados na roleta. A Figura 5 mostra uma representação do método de seleção por roleta.

Figura 5 - Exemplo de seleção por roleta.



Fonte: Próprio autor.

O pseudocódigo do método de seleção por roleta é mostrado detalhadamente na Tabela 4.

Tabela 4 - Pseudocódigo do método de seleção por roleta.

<p>Entrada: $\varphi_s(x_i(t))$ para $i = 0 \dots n - 1$</p> <p>Saída: Índice i do indivíduo selecionado</p> <p>Início</p> <p>$i = 0$</p> <p>$soma = \varphi_s(x_i(t))$</p> <p>$r = \text{rand}()$</p> <p>Enquanto $soma < r$ faça</p> <p>$i = i + 1$</p> <p>$soma = soma + \varphi_s(x_i(t))$</p> <p>Fim- enquanto</p> <p>Selecione indivíduo i</p> <p>Fim</p>
--

Fonte: Próprio autor.

3.1.3.4 Seleção por torneio

Esse processo consiste em selecionar um grupo de t indivíduos aleatoriamente, a partir da população total, para participar de uma competição entre os t selecionados para ver quem ganha o direito de gerar filhos. O indivíduo que possuir melhor *fitness* é escolhido dentre os demais e o processo é repetido até que se tenha uma nova população.

3.1.4 Cruzamento (Crossover)

A partir da população atual, uma nova é formada pelo cruzamento aleatório entre os cromossomos, nos quais os filhos recebem características genéticas de seus pais.

Desse cruzamento, espera-se que as boas características de prévias gerações sejam passadas às próximas.

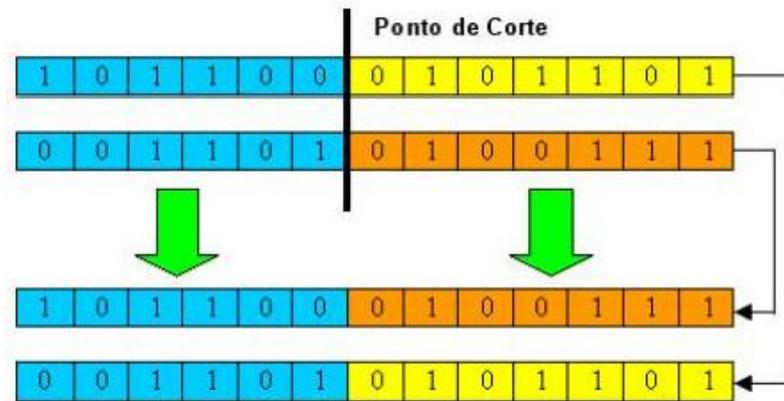
O cruzamento é um processo sexuado - envolve mais de um indivíduo - que emula o fenômeno de troca de fragmentos de genes entre pares de cromossomos, misturando características de dois indivíduos pais, simulando a reprodução de genes entre células. Durante o cruzamento, trechos de genes de um indivíduo são trocados por trechos equivalentes de outro pai. O resultado dessa operação é a formação de dois novos indivíduos contendo a combinação das melhores características de seus pais.

Existem muitos tipos de cruzamentos, como mais utilizados pode-se destacar nas subseções a seguir (KARAKATIČ; PODGORELEC, 2015) (MCPHEE et al., 2008):

3.1.4.1 Cruzamento de um ponto

Nesse tipo de cruzamento um ponto de corte aleatório é escolhido nos indivíduos selecionados. Na sequência é realizada uma troca cruzada do material genético dos pais, resultando em dois novos filhos. A Figura 6 ilustra a operação de cruzamento de um ponto, no qual o ponto de corte está entre o sexto e sétimo caractere. Como pode ser percebido ao analisar a figura, o material genético de cor azul foi mantido nos novos indivíduos e o conjunto de genes de cor amarela e laranja foram permutados.

Figura 6 - Exemplo de cruzamento de um ponto.

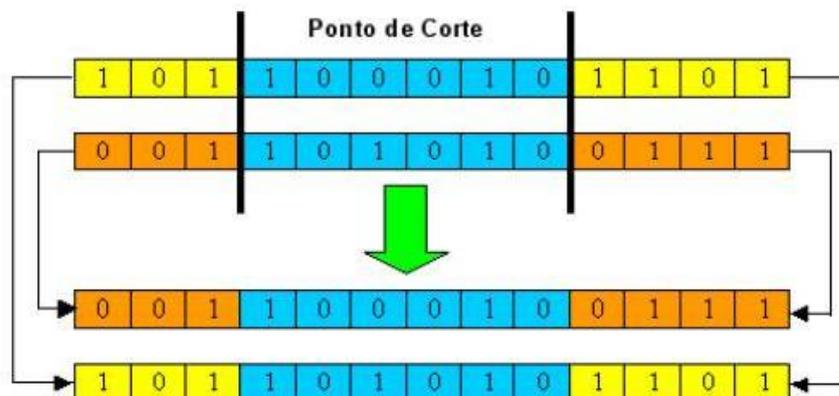


Fonte: Próprio autor.

3.1.4.2 Cruzamento de dois pontos

O procedimento é similar ao cruzamento de um ponto, com diferença que neste são selecionados dois pontos de cortes. Os conjuntos de genes a serem trocados podem ser das extremidades ou do meio dos indivíduos pais, dependendo do problema a ser resolvido. A Figura 7 apresenta um exemplo de cruzamento de dois pontos, onde somente o grupo de genes de cor azul foi mantido, enquanto os grupos de cor amarela e laranja das extremidades foram permutados.

Figura 7 - Exemplo de cruzamento de dois pontos.

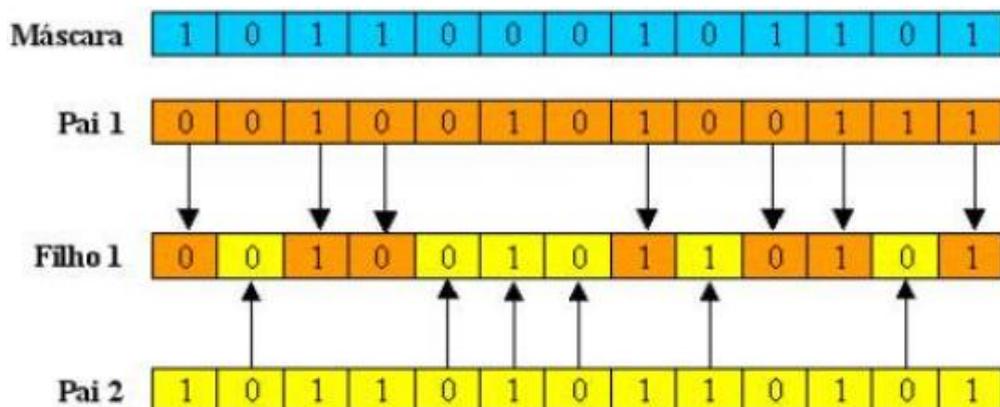


Fonte: Próprio autor.

3.1.4.3 Cruzamento uniforme

Esse cruzamento é bem diferente do cruzamento de um ponto e de dois pontos. Cada gene do descendente é gerado copiando o gene correspondente de um dos pais, escolhido de acordo com uma máscara de cruzamento, gerada aleatoriamente. Essa máscara é uma espécie de molde que tem como objetivo indicar quais genes serão copiados do primeiro pai e quais serão copiados do segundo. A Figura 8 mostra um exemplo do cruzamento uniforme. Na figura é possível verificar que os genes indicados pelo código 1 na máscara, o gene correspondente será copiado do Pai 1 para o Filho 1, caso contrário, onde houver 0, o gene correspondente será copiado do para o Pai 2.

Figura 8 - Exemplo de cruzamento uniforme.



Fonte: Próprio autor.

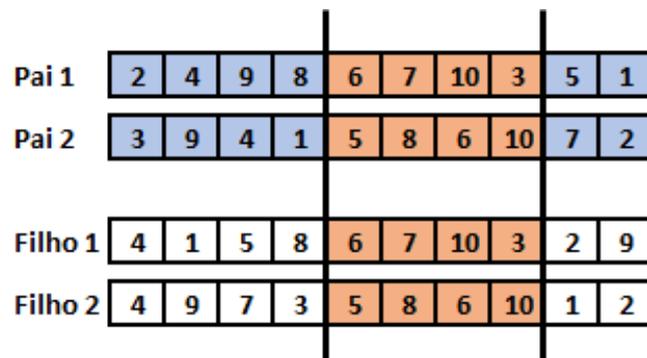
3.1.4.4 Cruzamento ordenado (*Order Crossover - OX*)

Este operador gera filhos a partir da escolha de uma sequência parcial de cidades de um dos pais e preservando a ordem relativa das cidades do outro pai. Ele trabalha sobre um subconjunto de cidades que ocupa um mesmo subconjunto de

posições em ambos os pais, escolhendo um subroteiro de um dos pais e preservando a ordem relativa das cidades de outro pai, como pode ser observado na Figura 9.

Os filhos Filho 1 e Filho 2 herdam as faixas entre os pontos de cortes, selecionados aleatoriamente, de seus respectivos pais. Na sequência, partindo do segundo ponto de corte de um pai (por exemplo *Pai 2*), copia-se as cidades na mesma ordem em que aparecem, removendo aquelas já contidas entre os dois pontos de cortes do outro pai (no caso *Pai 1*). Assim, a partir da sequência 7, 2, 3, 9, 4, 1, 5, 8, 6 e 10 (pertencente ao *Pai 2*), obtêm-se a 2, 9, 4, 1, 5 e 8 a ser inserida no Filho 1 a partir do seu segundo ponto de corte, removendo 6, 7, 10 e 3 já presentes em sua cadeia genética.

Figura 9 - Exemplo de cruzamento ordenado.



Fonte: Próprio autor.

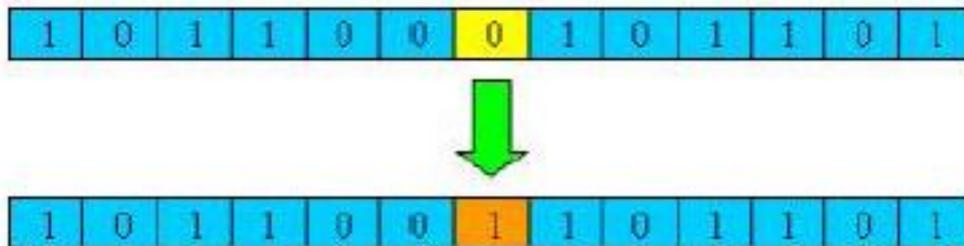
3.1.5 Mutação

A mutação é um método utilizado para a introdução e manutenção da diversidade genética da população que visa garantir uma maior varredura do espaço de busca e evitar que o AG convirja muito cedo para mínimos locais, pois promove alterações que direcionam a pesquisa para outras regiões do espaço de busca. Além disso,

aumenta a diversidade genética da população, evitando que ela entre em estagnação.

Após o cruzamento, o operador de mutação percorre toda a cadeia de genes do indivíduo, gerando um evento baseado na probabilidade de mutação. Se esse evento ocorrer, logo o operador de mutação é executado e o valor do gene correspondente é alterado. Se a alteração gerar indivíduos mais adaptados, essa nova característica será transmitida para os próximos indivíduos ao longo das gerações. No entanto, como na mutação genética, o operador de mutação deve ter uma taxa baixa, geralmente de 1%, ocasionando mudanças em poucos indivíduos, evitando que boas soluções sejam muito distorcidas. Por outro lado, cuidados devem ser tomados para evitar que a mutação gere indivíduos inválidos (KARAKATIČ, 2015). A Figura 10 mostra um exemplo de ocorrência de mutação em um indivíduo.

Figura 10 - Exemplo do operador de mutação.



Fonte: Próprio autor.

O operador de mutação pode ocorrer de duas maneiras: uniforme ou ordenada. Na mutação ordenada, a posição do gene é escolhida aleatoriamente e sofre uma mutação conforme a probabilidade P_m . Já no método ordenado, dois pontos da cadeia de genes do indivíduo são escolhidos aleatoriamente e somente os genes entre esses dois limites podem sofrer mutação (GOLDBERG, 1989). As Tabelas 5 e 6 apresentam os pseudocódigos de implementação desses dois métodos.

Tabela 5 - Pseudocódigo do método de mutação uniforme.

<p>Entrada: Novo indivíduo I_i Saída: Novo indivíduo I'_i após sofre mutação uniforme</p> <p>Início Para $j = 0$ até $n - 1$ faça Se $\text{rand}() \leq Pm$ então //complementa gene $I'_i(j) = \sim I_i(j)$ Fim-se Fim-para</p> <p>Fim</p>

Fonte: Próprio autor.

Tabela 6 - Pseudocódigo do método de mutação ordenada.

<p>Entrada: Novo indivíduo I_i Saída: Novo indivíduo I'_i após sofre mutação uniforme</p> <p>Início // Seleciona primeiro ponto de mutação entre 0 e $n - 1$ $t_1 = \text{rand}(0, n - 1)$ // Seleciona segundo ponto de mutação entre 0 e $n - 1$ $t_2 = \text{rand}(0, n - 1)$</p> <p> Para $j = t_1$ até t_2 faça Se $\text{rand}() \leq Pm$ então //complementa gene $I'_i(j) = \sim I_i(j)$ Fim-se Fim-para</p> <p>Fim</p>
--

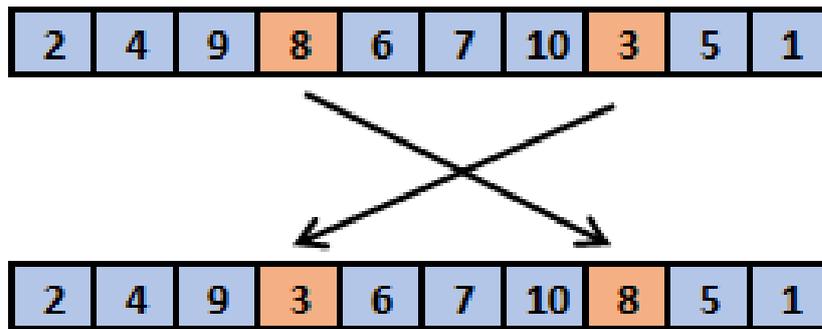
Fonte: Próprio autor.

3.1.5.1 Exchange Mutation

No operador de mutação por troca (*exchange mutation*) dois genes de um indivíduo são selecionados aleatoriamente sem qualquer restrição e suas posições são

trocadas. A Figura 11 mostra um exemplo desse operador de mutação, na qual o valor 8 da quarta posição foi trocado pelo valor 3 da oitava posição.

Figura 11 - Exemplo de *Exchange Mutation*.



Fonte: Próprio autor.

3.1.6 Critério de parada do algoritmo genético

Como os AGs buscam a otimização da solução, seria ideal que o algoritmo só parasse ao encontrar a melhor solução. Porém, não se pode afirmar com certeza se uma solução encontrada pelo algoritmo corresponde a um ótimo global. Assim, usa-se como critério de parada um número máximo de gerações ou tempo limite de processamento do algoritmo (MCPHEE; POLI, LANGDON, 2008).

Outro critério usado como condição de parada do algoritmo é a estagnação da população, ou seja, quando após algumas gerações não se observa melhoria na população e ela não mais evolui ficando estagnada sempre no mesmo nível de evolução (LINDEN, 2012).

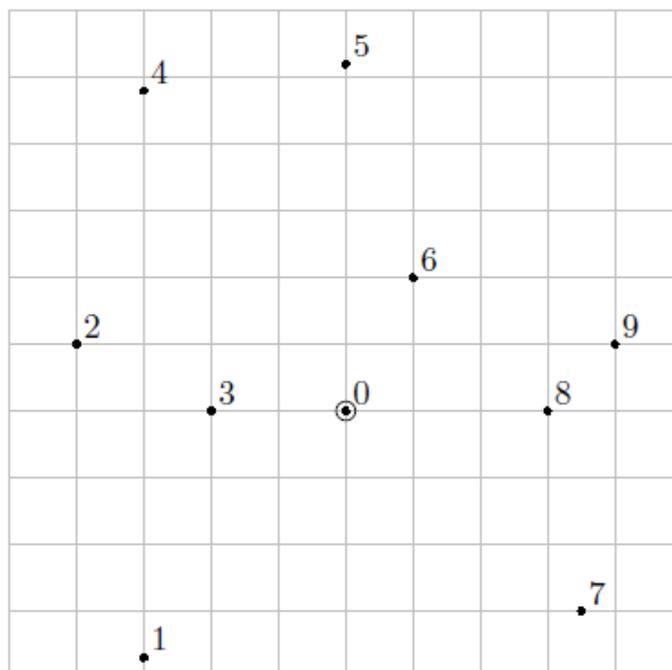
3.1.7 Aplicação do algoritmo genético em problemas de rota

Como mencionado em (BRAEKERS; RAMAEKERS, VAN NIEUWENHUYSE, 2016) e (KARAKATIČ; PODGORELEC, 2015), uma grande quantidade de problemas de rotas pode ser resolvida utilizando o Algoritmo Genético.

Na resolução do problema tradicional do caixeiro viajante, em que há uma lista de cidades a serem visitadas e cada cidade representa um ponto de parada obrigatória, em que a função objetivo é minimizar o custo de viagem entre cada um desses pontos. Dessa forma, sendo n o número de cidades, uma solução é dada por uma sequência de todas as n cidades a serem visitadas.

A Figura 12 mostra um exemplo de um espaço de busca, em que as cidades são apresentadas pelos números naturais consecutivos $0, 1, 2, \dots, n$. O ponto de partida e também de retorno é apresentado pelo número 0 .

Figura 12 - Mapas das cidades em um problema do caixeiro viajante.



Fonte: Próprio autor.

Para resolver esse problema também é necessário conhecer as distâncias entre todas as cidades. Considerando distância Euclidiana, essas são apresentadas na Tabela 7, mostrada a seguir.

Tabela 7 - Distâncias entre as cidades.

	0	1	2	3	4	5	6	7	8	9
0	0,00	4,76	4,12	2,00	5,66	5,20	2,24	4,61	3,00	4,12
1	4,76	0,00	4,81	3,83	8,50	9,39	6,96	6,54	7,05	8,43
2	4,12	4,81	0,00	2,24	3,93	5,80	5,10	8,50	7,07	8,00
3	2,00	3,83	2,24	0,00	4,90	5,57	3,61	6,26	5,00	6,08
4	5,66	8,50	3,93	4,90	0,00	3,03	4,88	10,15	7,68	7,96
5	5,20	9,39	5,80	5,57	3,03	0,00	3,35	8,92	6,00	5,80
6	2,24	6,96	5,10	3,61	4,88	3,35	0,00	5,59	2,83	3,16
7	4,61	6,54	8,50	6,26	10,15	8,92	5,59	0,00	3,04	4,03
8	3,00	7,05	7,07	5,00	7,68	6,00	2,83	3,04	0,00	1,41
9	4,12	8,43	8,00	6,08	7,96	5,80	3,16	4,03	1,41	0,00

Fonte: Próprio autor.

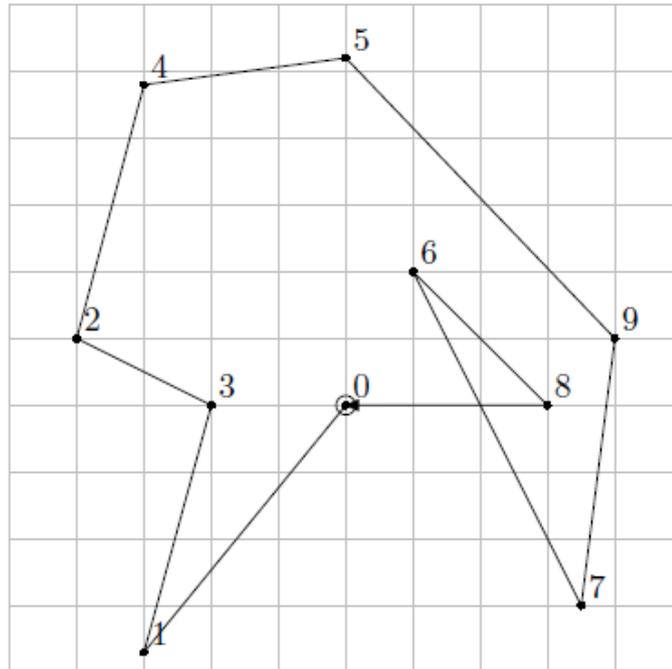
Ao utilizar o AG para resolver o problema apresentado é necessário definir a população inicial de indivíduos (cromossomos), sendo cada gene dos indivíduos formado por uma cidade existente no problema, como mostrado na Figura 13. Já a Figura 14 ilustra a rota gerada por esse indivíduo.

Figura 13 - Representação de indivíduo contendo uma solução parcial.

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

Fonte: Próprio autor.

Figura 14 - Representação da solução parcial do problema.



Fonte: Próprio autor.

A função de custo do problema para um determinado indivíduo (solução) é dada por meio da distância total percorrida pelo caixeiro viajante ao passar por cada cidade e retornar para a cidade de origem. O custo do indivíduo mostrado na Figura 12 é calculado somando as distâncias entre as cidades, como mostrado na Equação 3.6, a seguir:

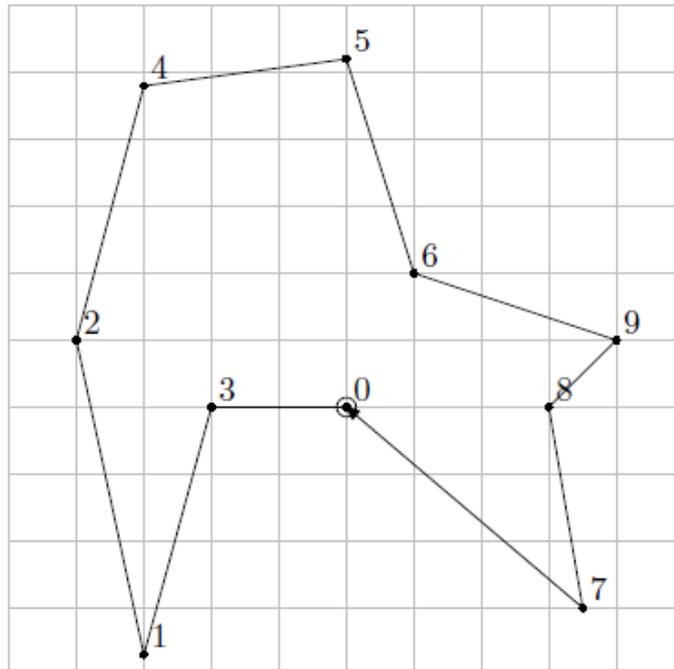
$$d(0,1) + d(1,3) + d(3,2) + d(2,4) + d(4,5) + d(5,9) + d(9,7) + d(7,8) + d(8,6) + d(6,9) + d(9,0) = 39,04 \quad (2.6)$$

onde $d(a, b)$ é a distância entre as cidades a e b .

Após definição da população inicial do algoritmo, são realizados os operadores genéticos: seleção dos indivíduos, cruzamento e mutação, que fazem com que a cada iteração do algoritmo uma melhor solução seja encontrada.

Quando o algoritmo alcança o critério de parada, o indivíduo que contém a rota de menor custo de deslocamento entre o conjunto de cidades é selecionado para representar a solução final do problema, que não é necessariamente ótima, mas sim a solução satisfatória (sub-ótima). A Figura 15 apresenta a solução final encontrada para o problema tradicional do caixeiro viajante, que tem o custo total de 33,17.

Figura 15 - Melhor solução encontrada para o problema.



Fonte: Próprio autor.

3.1.8 Pontos positivos na utilização do AG na resolução de problemas de geração de rotas automáticas

- Os Algoritmos Genéticos trabalham com base na codificação do problema e oferecem suporte para a elaboração de um programa computacional abrangente (RAIMANN, 2007);

- O Algoritmo Genético comumente encontra soluções de boa qualidade, sendo encontrada a solução ótima para alguns tipos de problemas/instâncias (GOLDBERG, 1989);
- Procuram a solução para o problema por meio de uma população de indivíduos e não em um ponto de cada vez;
- São robustos e podem resolver uma grande diversidade de problemas complexos de forma rápida e confiável (RAIMANN, 2007);
- São de fácil implementação e flexível para se adicionar modificações;
- São fáceis de combinar com outros métodos e algoritmos.

3.1.9 Pontos negativos na utilização do AG na resolução de problemas de geração de rotas automáticas

- Não há uma definição padrão do ponto de parada do algoritmo que atenda a todos os casos. Logo, o algoritmo pode realizar uma quantidade desnecessária de iterações. Ou até mesmo terminar antes que uma melhor solução seja alcançada;
- Por ser aleatória, a criação da população inicial do AG pode ser extremamente demorada e/ou custosa, podendo o algoritmo até mesmo não conseguir gerar o número necessários de indivíduos esperados para a execução das etapas seguintes do algoritmo, ficando o AG em repetições infinitas em busca de um número suficiente de indivíduos (MENDES et al., 2016);
- Requerem um grande número de avaliações de funções de aptidão;

- Qualquer alteração no espaço de busca implica no reprocessamento do algoritmo.

3.2 SPACE-FILLING CURVES

Curvas de preenchimento espacial, mais conhecida no termo em inglês *Space-Filling Curves* (SPC), são curvas planares contínuas e sem cruzamento de traços usadas com o objetivo de preenchimento de todo um espaço bidimensional, como um quadrado de lado unitário $[0,1] \times [0,1] \subset \mathbb{R}^2$ como descrito em Sagan (2012) e Bader (2012).

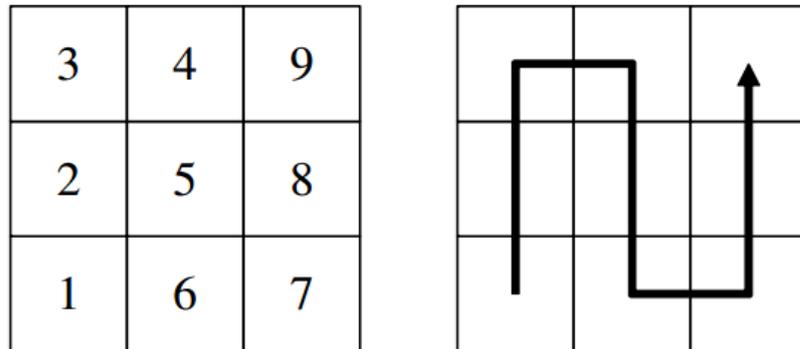
Inicialmente as SPC foram apresentadas pelo matemático italiano Giuseppe Peano (1890) e, atualmente, devido às suas propriedades de discretização espacial, são utilizadas em muitas aplicações práticas como em algoritmos de decomposição e até mesmo na resolução do problema do caixeiro viajante.

Inspirada nos padrões de crescimento existente na natureza, como a forma do crescimento das plantas e do crescimento celular, as SPC são formas fractais de preenchimentos de espaços constituídos por curvas contínuas tal que seus traços preenchem toda área bidimensional (como um quadrado) ou de N-dimensional (hipercubo) de uma maneira ordenada (BADER, 2012).

3.2.1 Curvas de Peano

A curva de Peano é construída a partir de um bloco básico gerado em um quadrado dividido em nove partes, ou seja, cada face é dividida em três partes, como mostra a Figura 16. A numeração mostrada indica a sequência pela qual as regiões do quadrado são percorridas.

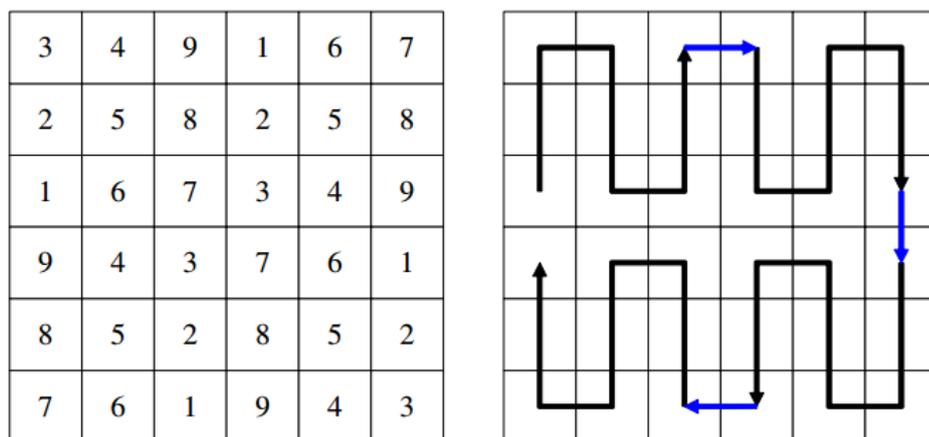
Figura 16 - Mapeamento e curva de Peano.



Fonte: Gorgulho Júnior (2017).

A divisão do quadrado em múltiplos de três permite a construção da curva pela repetição do padrão básico com pequena alteração na sequência. Na Figura 17 tem-se um quadrado com seus lados divididos em 6 partes. O padrão básico é apresentado em preto enquanto que os percursos de conexão estão destacados na cor azul.

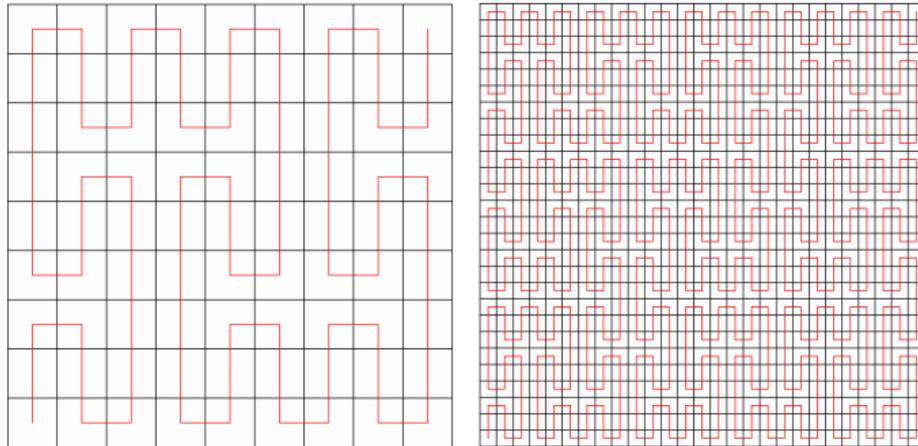
Figura 17 - Curva de Peano com 6 divisões por lado.



Fonte: Gorgulho Júnior (2017).

A Figura 18 apresenta a curva de Peano com o quadrado tendo suas faces divididas em 9 e 27 partes.

Figura 18 - Curvas de Peano com 9 e 27 divisões nas faces.



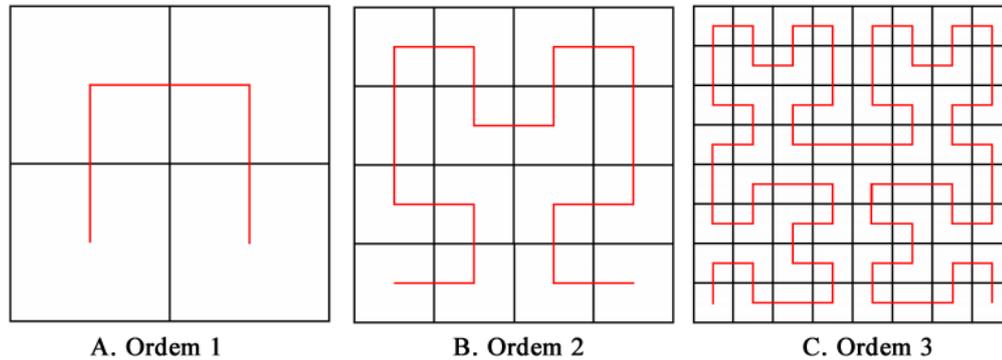
Fonte: Gorgulho Júnior (2017).

Embora Giuseppe Peano tenha apresentado pela primeira vez as curvas de preenchimento espacial (SERGEYEV; STRONGIN; LERA, 2013), foi o matemático David Hilbert (1900) que difundiu esse campo da geometria (BADER, 2012). A curva de Hilbert é uma curva de preenchimento do espaço que visita todos os pontos de uma grade quadrada com um tamanho de 2×2 , 4×4 , 8×8 , 16×16 ou qualquer outra potência de R^2 (SAGAN, 2012).

Os elementos básicos das curvas de Hilbert são chamados de "copos" (um quadrado com um lado aberto) e "junções" (um vetor que une dois copos). O lado "aberto" de um copo pode ser superior, inferior, esquerdo ou direito. Além disso, cada copo possui dois pontos finais, e cada um deles pode ser o ponto de "entrada" ou o ponto de "saída". Assim, existem oito variedades possíveis de copos. Na prática, uma curva de Hilbert usa apenas quatro tipos de copos e tem uma direção: para cima, para baixo, para a esquerda ou para a direita (BADER, 2012).

Como apresentado na Figura 19, uma curva de Hilbert de primeira ordem é apenas um único copo (Figura 19-A) que preenche um espaço 2×2 . Na curva de segunda ordem esse copo é substituído por quatro outros menores, que são ligadas por três junções (Figura 19-B). A cada próxima ordem o processo é repetido, substituindo cada copo por quatro outros menores interligados por três novas junções.

Figura 19 - Curva de Hilbert em diferentes ordens.

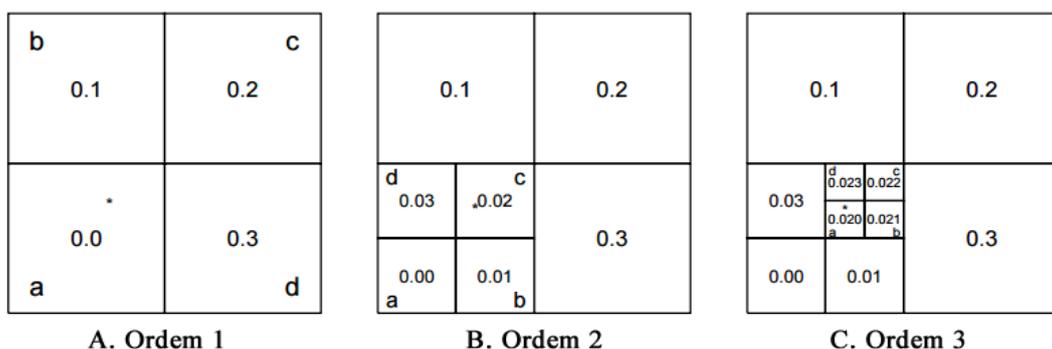


Fonte: Próprio autor.

3.2.2 Mapeamento da Curva de Hilbert

Inicialmente, as curvas de Hilbert impõem uma ordem de pontos a todas as células de sua matriz que são ordenados de acordo com a sequência em que a curva visita as células da matriz. Dessa forma, a classificação de um conjunto de pontos ocorre de acordo com a formação das curvas, que inicialmente precisa encontrar uma decomposição em que cada ponto esteja localizado em diferentes células para somente então estabelece a ordenação dos pontos (BARTHOLDI III; GOLDSMAN 2001a). A Figura 20 mostra em detalhes como é realizada a decomposição da Curva de Hilbert.

Figura 20 - Exemplo de decomposição da Curva de Hilbert.

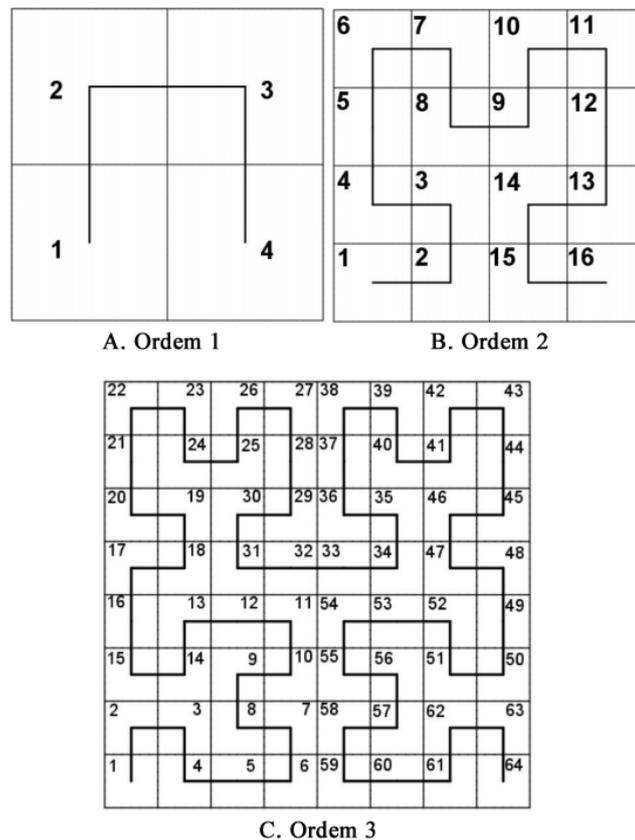


Fonte: Bartholdi III (2001a).

Após a formação completa das curvas no preenchimento de uma determinada área, uma ordem das células visitadas é gerada, consequentemente possibilitando o mapeamento completo da área em questão (ROSE, 2001).

A Figura 21 mostra como é formada a sequência de mapeamento de uma área utilizando as curvas de preenchimento de espaço proposto por Hilbert. É possível perceber ao analisar a figura que para cada nível de decomposição há um diferente mapeamento e quanto maior a ordem, mais detalhando é o nível do mapeamento.

Figura 21 - Sequência da ordem de mapeamento da Curva de Hilbert.



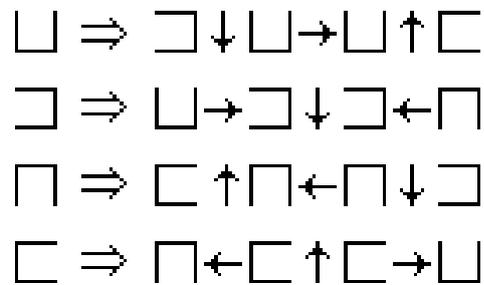
Fonte: Rose (2001).

3.2.3 Implementação do algoritmo das Curvas de Hilbert

Como falado anteriormente, as curvas de Hilbert são compostas por apenas quatro tipos de copos: superior, inferior, esquerdo ou direito. E em cada nova decomposição o copo atual é substituído por outros quadros menores.

A decomposição é sempre realizada quando há um interesse de aumentar o detalhamento da área mapeada. Ela deve ser realizada seguindo uma regra definida pelo algoritmo, que define a substituição dos copos conforme a sequência mostrada na Figura 22.

Figura 22 - Regras de subdivisão dos copos existentes nas curvas de Hilbert.



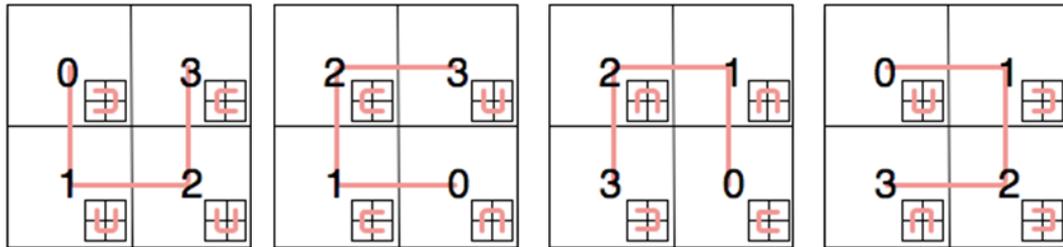
Fonte: Próprio autor.

Caso um copo tenha sua abertura voltada para cima (\sqcup), então o quadrante será dividido em outros quatros e preenchido por novos quadros que serão criados conforme a seguinte sequência:

- um copo com abertura para esquerda (\sqsubset) + uma junção para abaixo;
- um copo com abertura para cima (\sqcup) + uma junção para direita;
- um copo com abertura para cima (\sqcup) + uma junção para cima;
- um copo com abertura para direita (\sqcup);

A Figura 23 mostra detalhadamente como é realizada a substituição de um copo de um determinado nível para outro de maior ordem.

Figura 23 - Regra de decomposição de níveis.



Fonte: Próprio autor.

A Tabela 8 apresenta a implementação da função recursiva responsável pela geração das curvas de Hilbert.

Tabela 8 - Código de implementação das curvas de Hilbert.

```
function [x,y] = hilbert(n)

    % end of recursion
    if n <= 0
        x = 0;
        y = 0;

    % recursive method call
    else
        [x0,y0] = hilbert(n-1);

    %calculate new co-ordinates
        x = 0.5*[-0.5+y0 -0.5+x0 0.5+x0 0.5-y0];
        y = 0.5*[-0.5+x0 0.5+y0 0.5+y0 -0.5-x0];
    end
end

% Call the hilbert function
[x,y] = hilbert(3);
plot(x,y,'-');
```

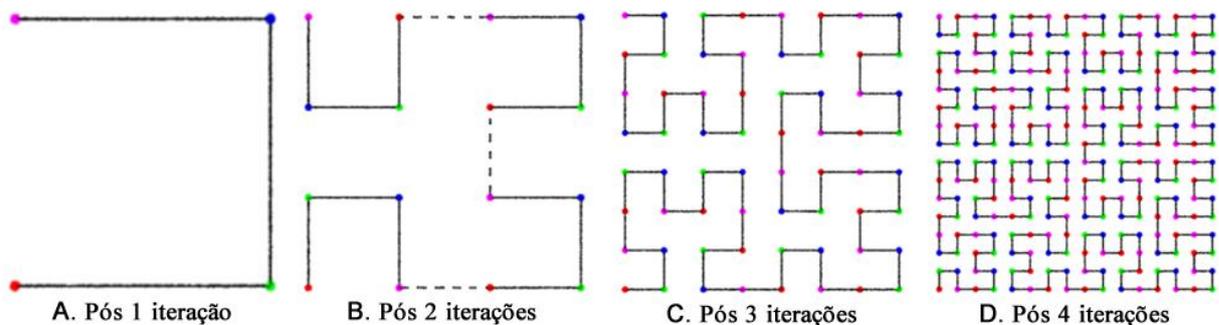
Fonte: Kesson (c2002).

O fragmento de código mostrado na Tabela 8 demonstra um exemplo de utilização da função responsável pela geração das curvas de Hilbert para o preenchimento de

espaços de uma área quadrada (R^2) seguindo as regras preestabelecidas pelo algoritmo.

Ao passar o parâmetro inicial de n , que representa o valor da ordem da curva de preenchimento, a função processa os valores de cada nível recursivamente, e retorna as coordenadas dos pontos x e y . A Figura 24 apresenta os resultados da função a cada iteração do algoritmo.

Figura 24 - Níveis de iteração da função de geração das curvas de Hilbert.



Fonte: Kesson (c2002).

Chris Muelder e Kwan-Liu Ma (2008) apresentam em seu trabalho o pseudocódigo para a implementação das curvas de Hilbert e propõe uma nova abordagem para geração de *layouts* gráficos através do uso de curvas de preenchimento espacial.

Em Kesson (c2002) são disponibilizados importantes detalhes da implementação das curvas de Hilbert, bem como de algumas variações e adaptações desse fractal. Em complemento, também são disponibilizados os códigos de implementação em diferentes linguagens de programação.

3.2.4 Aplicação da Curva de Hilbert em problemas de rota

Uma propriedade útil de uma curva de preenchimento de espaço é que ela tende a visitar todos os pontos de uma região depois de ter entrado em seu quadrante. Assim, pontos que estão próximos uns dos outros no plano tendem a estarem próximos uns dos outros ao longo da curva. Dessa forma, pontos espalhados ao longo de uma área podem ser visitados na mesma sequência da formação da curva de preenchimento de espaço.

Segundo Bartholdi III (2017b), o método das curvas no preenchimento tem sido usada em muitas empresas na construção de rotas de menor custo, como:

- *Meals-on-Wheels* em Fulton County (Atlanta, GA), que entrega diariamente centenas de refeições para pessoas doentes ou idosas;
- Para encaminhar entrega de sangue pela Cruz Vermelha Americana para hospitais na área metropolitana de Atlanta;
- Na geração de rotas de visitas aos pontos turísticos de cidades da Alemanha.

Para aplicar a curva de preenchimento proposta por Hilbert no problema de geração de rotas o algoritmo segue os seguintes passos:

- a. Os pontos a serem visitados são georreferenciados no sistema GIS da CESAN;
- b. Os pontos são enquadrados nas células da matriz gerada pelo algoritmo, de forma que cada ponto fique em uma célula. A ordem do fractal é ajustada para possibilitar o encaixe de cada ponto.
- c. Por fim, a rota é gerada conforme a sequência de mapeamento gerada pela Curva de Hilbert.

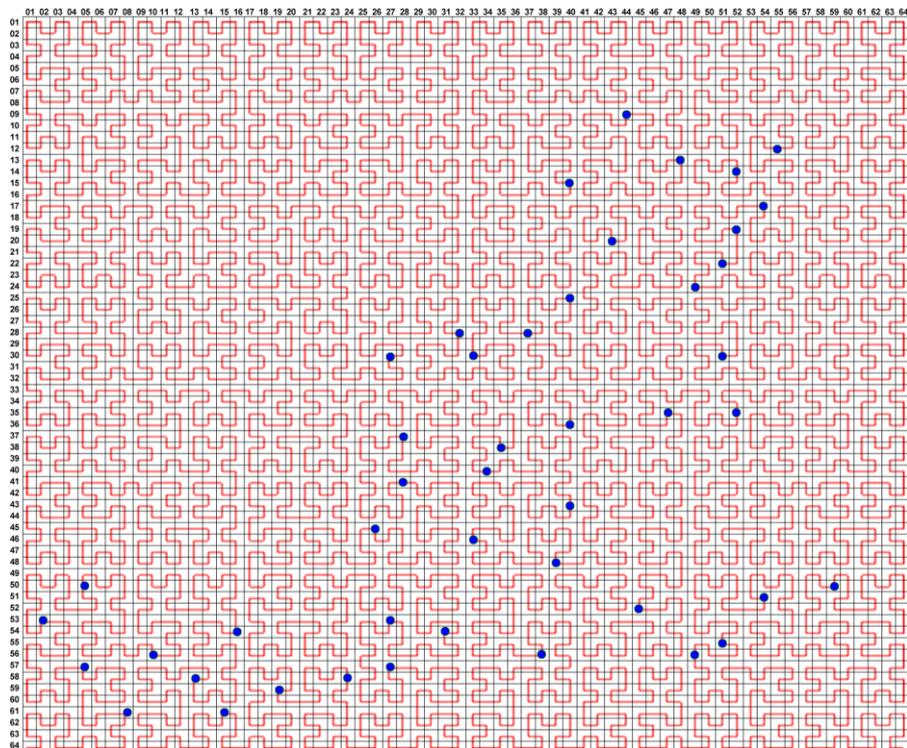
As Figuras 25, 26 e 27 apresentam os passos listados anteriormente para a geração de uma rota de atendimento de Solicitação de Serviços no município de Vitória contendo 46 pontos. O resultado do algoritmo gerou uma rota com custo de 28.740,26 metros.

Figura 25 - Pontos georreferenciados no sistema GIS.



Fonte: Próprio autor.

Figura 26 - Pontos encaixados na matriz de mapeamento da Curva de Hilbert de ordem 6.



Fonte: Próprio autor.

Figura 27 - Rota gerada por meio da sequência da ordem de mapeamento da Curva de Hilbert.



Fonte: Próprio autor.

3.2.5 Pontos negativos na utilização da Curva de Hilbert na resolução de problemas de geração de rotas automáticas

- O algoritmo é aplicado somente na geração de rotas abertas, isto é, em rotas que não há uma obrigatoriedade de o veículo retornar para uma base, depósito ou ponto de partida;
- O algoritmo não leva em consideração as restrições existentes em problemas de rotas, como tempo de percurso ou tempo de permanência em cada cliente. Ele gera apenas um sequenciamento dos pontos a serem visitados;

- A rota gerada pelo algoritmo fractal leva em consideração somente a distribuição dos pontos no plano e não considera a distância (custo) de deslocamento entre cada um dos pontos contidos no espaço de busca;
- O algoritmo não pode ser aplicado em geração de rotas em que não há obrigatoriedade de visita a todos os pontos do espaço de busca;
- O algoritmo não garante gerar a melhor rota possível para todos os problemas de rotas;
- Pontos indesejados precisam ser removidos do espaço de busca antes da execução do algoritmo.

3.2.6 Pontos positivos na utilização da Curva de Hilbert na resolução de problemas de geração de rotas automáticas

- O algoritmo garante que todos os pontos estarão na rota gerada, sem que haja caminhos redundantes;
- O algoritmo pode ser aplicado em casos dinâmicos, em que novos pontos são adicionados ou removidos do espaço de busca sem que haja a necessidade de realizar ajustes na implementação do algoritmo;
- Algoritmo é de fácil implementação;
- O algoritmo é rápido, necessário apenas $O(n \log n)$ de esforço para construir um caminho de n pontos e apenas $O(\log n)$ de esforço para atualizar a solução, adicionando ou removendo pontos (BARTHOLDI III, 2017a).

4 RESTRIÇÕES, CATEGORIZAÇÃO E MODELO MATEMÁTICO

Neste capítulo serão apresentadas as restrições existentes no problema de roteamento da empresa CESAN. Baseado nessas restrições foi formulado a categorização do problema de rota e sua respectiva formulação matemática.

4.1 CRITÉRIOS DO PROBLEMA PROPOSTO

Após a realização de uma série de verificações em campo e entrevistas realizadas com os empregados encarregados pelo planejamento de rotas, as seguintes restrições para o problema de roteamento de veículos da CESAN foram definidas:

- a. Cada veículo atende uma única rota por vez;
- b. As rotas, obrigatoriamente, começam e terminam em uma unidade da CESAN;
- c. Um veículo, ao terminar o seu percurso, deve retornar para a sua unidade de origem;
- d. Os veículos atendem somente clientes localizados no município de sua unidade;
- e. Cada ponto do percurso deve ser atendido uma única vez e por um único veículo;
- f. A empresa disponibiliza um total de 16 veículos para cada município da região metropolitana;
- g. Ao realizar o seu trabalho, o empregado deve chegar à cada ponto contido na rota em um tempo máximo de 15 minutos;

- h. O tempo de realização do serviço para inspecionar cada ponto deve ser no máximo de 15 minutos;
- i. Ao longo de um dia, um empregado percorre uma rota contendo no máximo 16 e no mínimo 14 pontos de inspeção;
- j. O atendimento a um conjunto inferior a 14 pontos deve ser realizado no dia seguinte, quando novos pontos deverão surgir, possibilitando a criação de rotas contendo o número máximo e mínimo de pontos, respeitando os critérios estabelecidos anteriormente.

4.2 CATEGORIZAÇÃO DO PROBLEMA DE ROTA

A seguir estão detalhadas as principais características do problema de rota da empresa, identificadas por meio das restrições:

- Tipos de veículos disponíveis: motos;
- Limite de capacidade dos veículos: não se aplica;
- Horário de início das viagens: 08:00;
- Horário de término: 17:00;
- Tipo de operação: inspeção dos pontos determinados, podendo ser em casas, edifícios, comércios, ruas, praças, parques, rios e praias;
- Tamanho da frota: limitada em 64 motos para os municípios da região da grande Vitória, sendo 16 para o município de Vitória, 16 para Serra, 16 para Vila Velha e 16 para Cariacica;
- Tipo de frota: homogênea;

- Número de Depósito: um para cada município;
- Visita e atendimento aos clientes: cada veículo realiza inspeção somente no município onde está localizado o seu depósito de origem. Logo, um veículo não pode sair dos limites do seu município para realizar qualquer tipo de atendimento;
- Tipo de demanda: determinística;
- Estrutura da rede: direcionada;
- Horizonte de planejamento das rotas: curto prazo.

4.2.1 Tipo de problema de rota

Uma vez que os veículos responsáveis pela realização das rotas não possuem restrições de capacidade, o problema de roteamento da CESAN é categorizado como um *Single Depot VRP (SDVRP)* que consiste em determinar múltiplos roteiros com o menor custo possível, de modo que cada veículo deva visitar um conjunto de pontos existentes em uma rede, e cada nó deve ser visitado uma única vez. Apesar da empresa CESAN possuir múltiplos depósitos, cada um está localizado em apenas um município, com isso, cada veículo deve retornar e sair da mesma base, não podendo ultrapassar os limites de seu município de atendimento.

4.3 FORMULAÇÃO MATEMÁTICA

A proposta de formulação matemática é apresentada apenas para a formalização para o problema de roteamento descrita a seguir é baseada em (NECULA; BREABAN; RASCHIP, 2015a, 2015b), e adota o padrão sintático comumente

utilizado na literatura. Para o problema de roteamento, será utilizada a representação de um grafo orientado, $G(N, E)$, em que N é o conjunto de nós, sendo cada nó um cliente, com o número máximo de clientes $|N|$, e E é o conjunto de caminhos entre cada cliente e a garagem, com $N = C \cup \{0, n + 1\}$ e $C = \{1, \dots, n\}$, em que C representa o conjunto de clientes e 0 e $n + 1$ são os nós de garagem.

Notação:

k : Índice de identificação do veículo, sendo v o número máximo de veículos disponíveis.

i, j : representam os clientes, assim $i, j \in N$. D_{ij} : Representa a distância do cliente i para j .

H_I : Representa o horário de início do roteamento.

H_F : Representa o horário de término do roteamento.

Parâmetros:

$T[i][j]$: Matriz de custo entre os clientes e a garagem. Nesta modelagem, usa-se o tempo de deslocamento entre um cliente a outro.

D : Tempo de atendimento de cada cliente.

Variáveis:

C^k : Indica o uso do veículo, assumindo 1 se o veículo k é usada e zero caso contrário.

$Y_{i,j}^k$: Variável binária, assumindo 1 se o veículo k passa pela rota que liga os nós i e j , e 0 caso contrário.

B_i^k : Indica o momento em que o veículo k chega ao nó i .

Função objetivo:

Como apresentado em (BEKTAS, 2006; NECULA; BREABAN; RASCHIP, 2015a; 2015b), nesta etapa são usadas 2 funções objetivos. Primeiro busca-se minimizar o número de veículos usados considerando que todos os clientes devem ser atendidos, Expressão 4.1, representando o custo dominante no problema. Na Expressão 4.2 busca-se minimizar a distância total percorrida por cada veículo.

$$\text{Min: } \sum_k^v C^k \quad (3.1)$$

$$\text{Min: } \sum_{k=1}^n \sum_{i=1}^n \sum_{j=1}^n T[i][j] \times Y_{i,j}^k \quad (3.2)$$

Restrições:

$$\sum_j^n Y_{0,j}^k = C^k; k = 1, \dots, v \quad (3.3)$$

$$\sum_i^n Y_{i,n+1}^k = C^k; k = 1, \dots, v \quad (3.4)$$

$$\sum_j^n Y_{i,j}^k - \sum_j^n Y_{j,i}^k = 0; i = 1, \dots, n; k = 1, \dots, v \quad (3.5)$$

$$\sum_k^v \sum_j^n Y_{i,j}^k = 1; i = 1, \dots, n \quad (3.6)$$

$$B_0^k = H_I; k = 1, \dots, v \quad (3.7)$$

$$B_n^k \leq H_F; k = 1, \dots, v \quad (3.8)$$

$$Se (Y_{i,j}^k = 1) \Rightarrow (B_j^k \geq B_i^k + D + T[i][j]) \quad (3.9)$$

Descrição:

As restrições de 4.3 a 4.9 modelam o problema do roteamento de veículos proposto por este trabalho. As Restrições 4.3 e 4.4 garantem que só receberão uma rota os veículos que serão usados no atendimento aos clientes. Já a Restrição 4.5 garante que todos os veículos que saem para realizar o atendimento aos clientes retornarão ao depósito de origem. A Restrição 4.6 garante que todos os clientes serão visitados. A Restrição 4.7 indica o horário de início do atendimento, enquanto que a Restrição 4.8 indica o horário de término. Por fim, a Restrição 4.9 garante o encadeamento dos horários de atendimento de cada cliente, isto é, o atendimento ao cliente localizado no nó j (B_j^1) deve ser maior ou igual ao atendimento do cliente

no nó i (B_i^1) mais o tempo para de atendimento do nó i (D) mais o tempo para percorrer o enlace ij ($T[i][j]$), que apresenta o tempo de deslocamento entre o primeiro e segundo cliente, se o veículo k usar o enlace ij ($Y_{i,j}^k$). Como esta restrição trata do encadeamento dos veículos no tempo, ela também trata o problema do *subtour* encontrado nos problemas de roteamento.

5 ALGORITMO HÍBRIDO PROPOSTO, RESULTADOS, COMPARAÇÕES E DISCUSSÕES

Neste capítulo é apresentado o algoritmo híbrido proposto por este trabalho e seus respectivos resultados. Além disso, é realizada uma análise comparativa entre os resultados alcançados pelo algoritmo híbrido e os resultados alcançados por soluções ótimas, disponibilizadas pela biblioteca de *benchmark* de otimização de VRPs, criada como parte do estudo publicado pelos autores em (UCHOA et al., 2017).

É importante destacar também que todos os resultados foram gerados utilizando-se uma máquina contendo o sistema operacional Windows Server 2012, com 8 núcleos do processador Intel Core I5 de 2.60 GHz, 16 GB de memória RAM e HD de 500 GB.

5.1 APLICAÇÃO DO AG NA RESOLUÇÃO DE UM PROBLEMA PROPOSTO

Inicialmente foi utilizada somente a implementação padrão AG na resolução de problemas de roteamento de veículos, seguindo as orientações propostas pelo artigo (KARAKATIČ, 2015), em que os parâmetros do algoritmo foram configurados com os seguintes valores:

- Uma população formada por 100 indivíduos em cada geração;
- Toda a população inicial é gerada aleatoriamente;
- Em cada geração, são selecionados pelo método de Elitismo 20 indivíduos de menor custo para participar na nova geração e para realizar cruzamento;
- Também são selecionados pelo método de Roleta outros 30 indivíduos, que também participarão na nova geração e irão realizar cruzamento;

- É realizado o cruzamento dos 50 indivíduos selecionados anteriormente, gerando 50 novos indivíduos, que farão parte da nova geração;
- O cruzamento é realizado pelo método de *Ordered Crossover*, conforme detalhado na seção 3.1.4.4;
- Após o cruzamento, cada indivíduo possui uma taxa de 25% de probabilidade de sofrer uma mutação, sendo o *Exchange Mutation* o método utilizado neste ponto, conforme detalhado em (KARAKATIČ; PODGORELEC, 2015);
- O ponto de parada do algoritmo é realizado na iteração de número 100 do algoritmo.

Após a parametrização, o algoritmo foi testado em 10 instâncias existentes na base de *benchmark* de otimização adaptadas para o SDVRP disponibilizadas por (UCHOA et al., 2017), que são apresentadas a seguir:

- A-n32-k5: formada por 1 depósito, 31 clientes e 5 veículos;
- A-n37-k5: formada por 1 depósito, 36 cidades e 5 veículos;
- CMT1: formada por 1 depósito, 49 cidades e 5 veículos.
- CMT3: formada por 1 depósito, 99 cidades e 8 veículos.
- CMT6: formada por 1 depósito, 49 cidades e 6 veículos.
- CMT8: formada por 1 depósito, 98 cidades e 9 veículos.
- P-n101-k4: formada por 1 depósito, 100 cidades e 4 veículos.
- X-n115-k10: formada por 1 depósito, 114 cidades e 10 veículos.
- X-n139-k10: formada por 1 depósito, 138 cidades e 10 veículos.
- X-n256-k16: formada por 1 depósito, 255 cidades e 16 veículos.

Para cada instância, o algoritmo foi executado 10 vezes com o objetivo de averiguar a confiabilidade dos resultados. A solução final de cada problema é dada pelo somatório de todas as rotas encontradas, sendo cada rota representada pelo melhor indivíduo da última geração do algoritmo. A Tabela 9 apresenta uma comparação entre dez problemas existentes na base de *benchmark*, que em relação à solução ótima, o algoritmo alcançou um resultado médio maior em 27,7%.

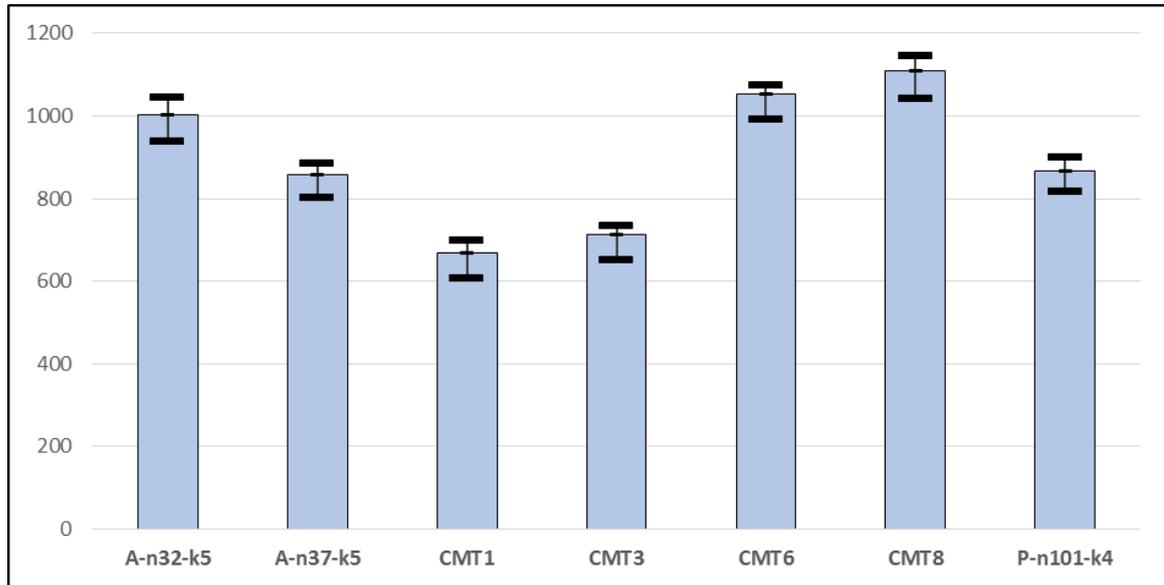
Tabela 9 - Comparação dos resultados alcançados pelo AG em relação como os resultados ótimos da base de *benchmark*.

Instância	Nº de Clientes + 1 depósito	Nº de Veículos	Solução Ótima	Algoritmo Genético impl. padrão		
			Custo	Custo Médio	Diferença entre os Custos	Diferença em %
A-n32-k5	32	5	784	1003	219	27,9%
A-n37-k5	37	5	669	857	188	28,1%
CMT1	50	5	524	667	143	27,3%
CMT3	100	8	555	712	157	28,3%
CMT6	50	6	826	1053	227	27,5%
CMT8	100	9	865	1109	244	28,2%
P-n101-k4	101	4	681	867	186	27,3%
X-n115-k10	115	10	12747	16125	3378	26,5%
X-n139-k10	139	10	13590	17292	3702	27,2%
X-n256-k16	256	16	18839	24156	5317	28,2%
Média:						27,7%

Fonte: Próprio autor.

Já o gráfico da Figura 28 apresenta as médias dos resultados das execuções de 7 das 10 instancias testadas e apresentadas na tabela anterior.

Figura 28 - Médias dos resultados alcançados pelo AG em sete instâncias contidas na base de *benchmark*.



Fonte: Próprio autor.

Na sequência, com propósito de melhorar os resultados alcançados pelo AG, uma operação de rearranjo genético foi adicionada ao algoritmo, após a etapa de mutação. Além disso, uma forma diferenciada de cruzamento e mutação também foi adicionada. Essas melhorias e seus resultados são detalhados das seções seguintes.

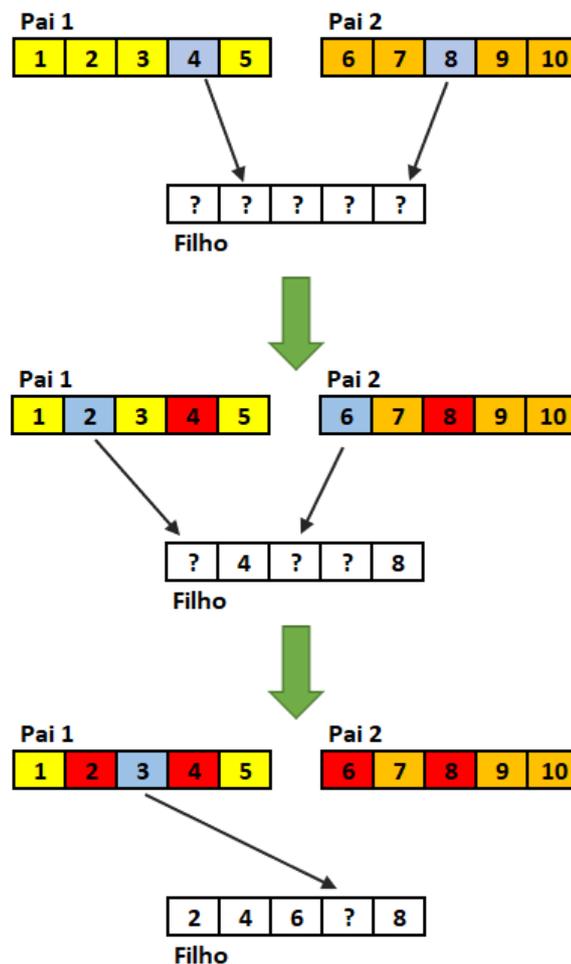
5.1.1 Cruzamento totalmente aleatório (*Random Crossover*)

Com o objetivo de aumentar a diversidade genética dos indivíduos filhos, a proposta desse cruzamento é que cada indivíduo filho seja constituído com 50% dos genes de cada pai, sendo esses genes selecionados aleatoriamente e também posicionados aleatoriamente no vetor genético do filho, tendo como restrição a inexistência de genes repetidos na cadeia genética do novo indivíduo.

Nesse cruzamento, enquanto o vetor genético do filho possui posições não preenchidas, os valores são escolhidos aleatoriamente de cada pai de forma alternada e posicionadas nos espaços vazios do vetor genético, selecionados também aleatoriamente. Na medida em que os valores dos vetores genéticos dos pais são selecionados, eles se tornam indisponíveis, não podendo ser novamente selecionado.

A Figura 29 apresenta um exemplo desse cruzamento, em que é possível observar que os genes selecionados dos pais são representados pela cor azul, enquanto que os genes de cor vermelha representam os genes já selecionados e indisponíveis para uma próxima seleção. As setas verdes representam as iterações das etapas de seleção dos genes.

Figura 29 - Exemplo de cruzamento totalmente aleatório.

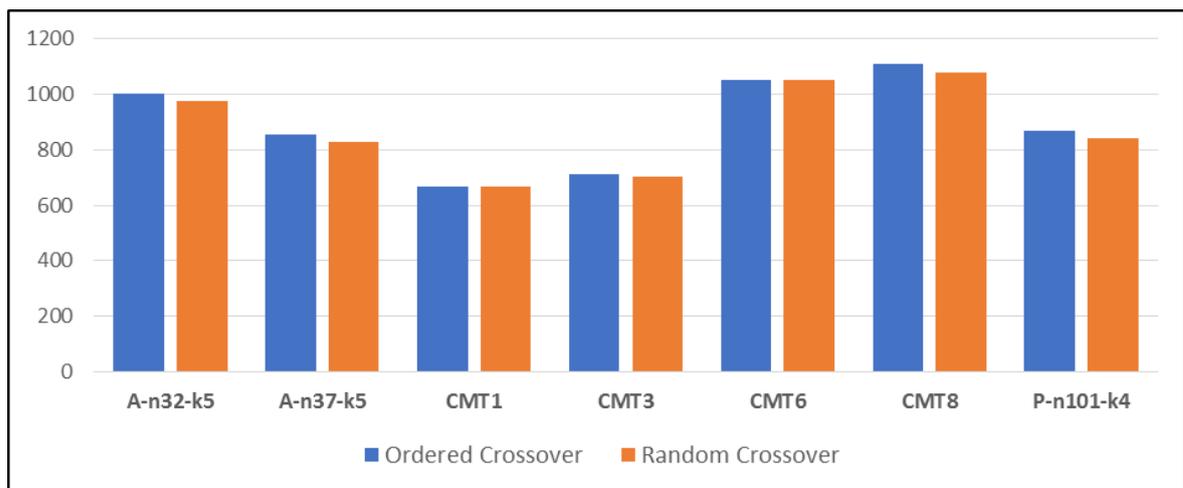


Fonte: Próprio autor.

Com intuito de validar a solução proposta, após a implementação desse operador de cruzamento, ele foi testado em 10 execuções consecutivas para cada uma das 10 instâncias existentes na base de *benchmark* de otimização, já citadas anteriormente, e seus resultados alcançados foram comparados com o cruzamento ordenado (*Ordered Crossover*), utilizado pelo artigo (KARAKATIČ, 2015). O gráfico da Figura 30 apresenta o resultado da comparação de 7 instâncias.

Como é possível verificar ao analisar o gráfico, o cruzamento totalmente aleatório foi ligeiramente melhor do que o cruzamento ordenado. Entretanto, em alguns casos, os dois cruzamentos atingiram os mesmos resultados (CMT1 e CMT6).

Figura 30 - Comparação dos resultados entre os cruzamentos ordenado e totalmente aleatório.



Fonte: Próprio autor.

5.1.2 Operação de rearranjo genético

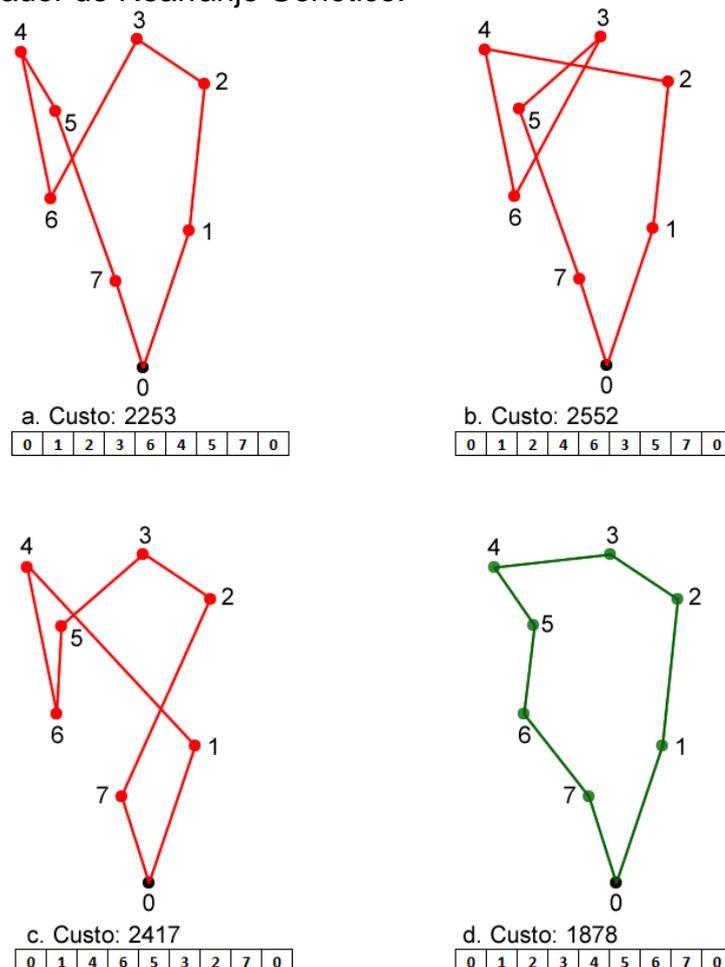
Devido à flexibilidade do AG, após a realização da fase de mutação do AG foi adicionado uma nova operação ao algoritmo, denominado de rearranjo genético, proposto neste trabalho. Essa nova operação consiste em verificar se o deslocamento para esquerda ou direita de um determinado gene diminui o custo de

uma solução local, sem a necessidade de mudança da carga genética do respectivo indivíduo.

Com esse operador, além de diminuir o custo da solução local, espera-se também que sejam eliminadas as sobreposições e cruzamentos dos trechos existentes na rota, criando, dessa forma, rotas circulares e, conseqüentemente, mais otimizadas.

A Figura 31 mostra alguns exemplos de rotas contendo sobreposições de trajeto e, na cor verde, o resultado final da rota após a utilização do operador de rearranjo genético. Como pode ser verificado (Figura 31-d), a rotina foi capaz de remover as sobreposições existentes e de reduzir consideravelmente o custo da solução realizando apenas uma permuta entre os genes, sem a necessidade de adicionar novos ou remover os existentes.

Figura 31 - Operador de Rearranjo Genético.

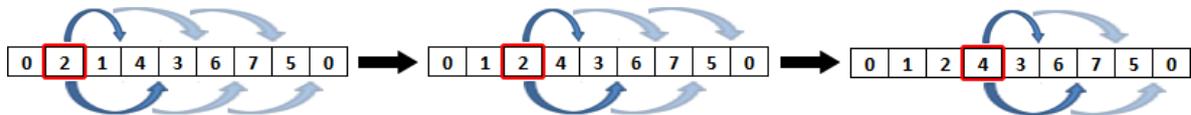


Fonte: Próprio autor.

Esse operador inicialmente verifica se o custo da solução é diminuído ao se deslocar para direita os genes posteriores ao do depósito. Caso o custo diminua, o gene é deslocado para a posição da verificação e o processo de checagem é reiniciado e repetido novamente, até que não haja nova redução no custo da solução. Em seguida uma nova verificação da direita para a esquerda é realizada para cada gene, chegando se o custo da solução diminui ao deslocar os genes. As Figuras 32 e 33 mostram o processo de deslocamento do gene, para a verificação da possível diminuição do custo da solução local.

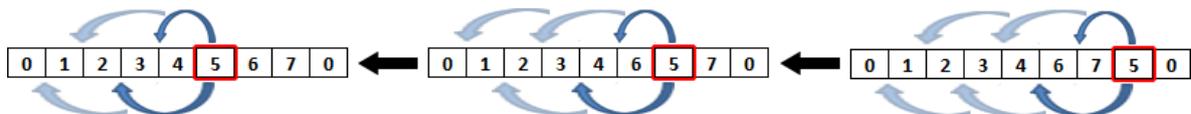
É importante ressaltar que esse processo é realizado sem levar em consideração os genes iniciais e finais, que representam o depósito. Ficando eles sempre fixos no início e fim da cadeia genética de cada indivíduo.

Figura 32 - Processo de deslocamento do gene da esquerda para a direita.



Fonte: Próprio autor.

Figura 33 - Processo de deslocamento do gene da direita para a esquerda.



Fonte: Próprio autor.

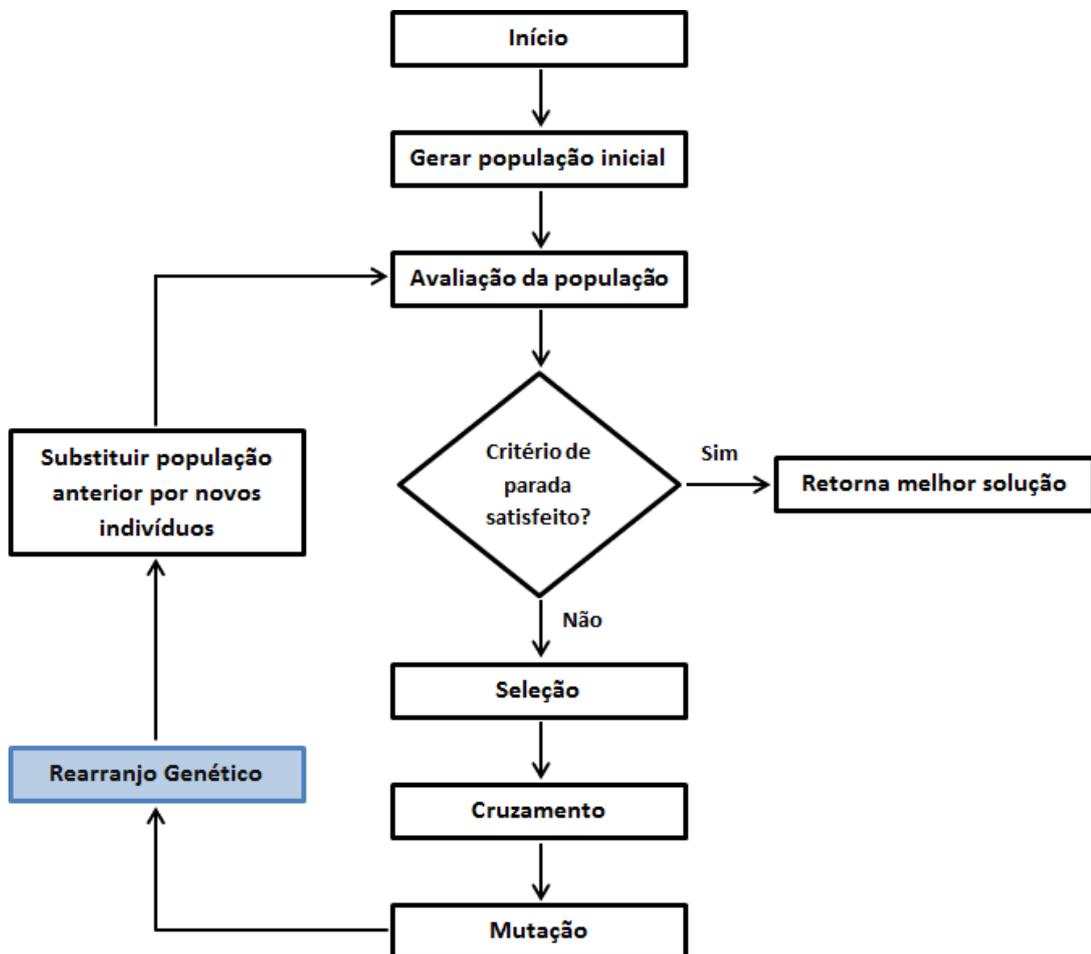
Após a inclusão desse operador ao AG, não houve a necessidade da realização da mutação por *Exchange Mutation*, que realiza a troca de posição de dois genes selecionados aleatoriamente de um indivíduo, devido ao fato do operador de Rearranjo Genético já realizar a permutação dos genes.

Por esse motivo, foi necessário realizar uma alteração no operador de mutação *Exchange Mutation*, que ao invés de apenas trocar a posição entre dois genes,

passou a trocar um gene selecionado aleatoriamente do indivíduo por outro existente no espaço de busca, também selecionado aleatoriamente.

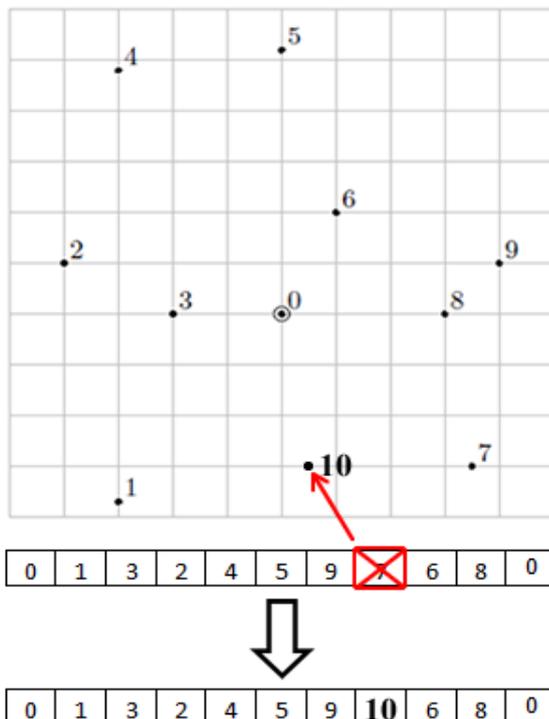
A Figura 34 apresenta a nova estrutura do algoritmo após a inclusão do operador de rearranjo genético, representado pela cor azul na imagem. Já Figura 5.8 apresenta um exemplo de como o operador de mutação passou a ser realizado.

Figura 34 - Estrutura do AG com o operador de Rearranjo Genético.



Fonte: Próprio autor.

Figura 35 - Representação da mutação por *Exchange Mutation* após alteração.



Fonte: Próprio autor.

Após adicionar todas as melhorias, o algoritmo foi novamente executado 10 vezes nos mesmos problemas existentes na base de *benchmark* apresentando por (UCHOA et al., 2017), com intuito comparar os resultados alcançados pelo AG com sua implementação padrão e o AG após a adição do operador do Rearranjo Genético e da mutação por *Exchange Mutation* alterada. A Tabela 10 apresenta uma comparação entre a implementação padrão AG com a versão contendo todas as melhorias (cruzamento totalmente aleatório, rearranjo genético e mutação). É possível observar que o algoritmo contendo as melhorias obteve uma melhora significativa de 9,6%, em relação ao AG padrão, e alcançou um resultado médio de 18,1% em relação à solução ótima.

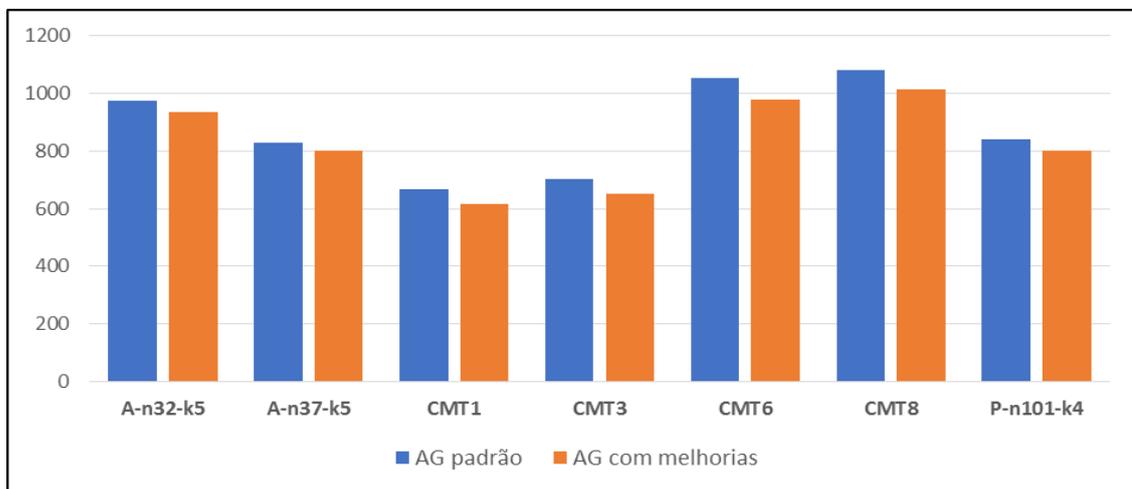
Tabela 10 - Comparação dos resultados alcançados pelo AG contendo melhorias e sua implementação padrão.

Instância	Nº de Clientes + 1 depósito	Nº de Veículos	Solução Ótima		Algoritmo Genético com melhorias				Algoritmo Genético impl. padrão			
			Custo	Custo Médio	Diferença entre os Custos	Diferença em %	Tempo de Exec. (s)	Custo Médio	Diferença entre os Custos	Diferença em %	Tempo de Exec. (s)	
A-n32-k5	32	5	784	937	153	19,5%	316	1003	219	27,9%	260	
A-n37-k5	37	5	669	802	133	19,9%	359	857	188	28,1%	277	
CMT1	50	5	524	615	91	17,4%	442	667	143	27,3%	318	
CMT3	100	8	555	653	98	17,7%	527	712	157	28,3%	375	
CMT6	50	6	826	978	152	18,4%	378	1053	227	27,5%	286	
CMT8	100	9	865	1014	149	17,2%	447	1109	244	28,2%	353	
P-n101-k4	101	4	681	803	122	17,9%	441	867	186	27,3%	354	
X-n115-k10	115	10	12747	15102	2355	18,5%	496	16125	3378	26,5%	383	
X-n139-k10	139	10	13590	16017	2427	17,9%	514	17292	3702	27,2%	392	
X-n256-k16	256	16	18839	22022	3183	16,9%	567	24156	5317	28,2%	413	
					Média:	18,1%			Média:	27,7%		

Fonte: Próprio autor.

Já o gráfico da Figura 36 mostra os resultados da comparação de 7 das 10 instancias testadas e apresentadas na tabela anterior. Ao analisar o gráfico é possível perceber a melhoria significativa que o operador de rearranjo genético e de mutação agregou ao algoritmo AG.

Figura 36 - Comparação do resultado da solução ao adicionar às melhorias ao AG.



Fonte: Próprio autor.

5.2 ALGORITMO HÍBRIDO PROPOSTO

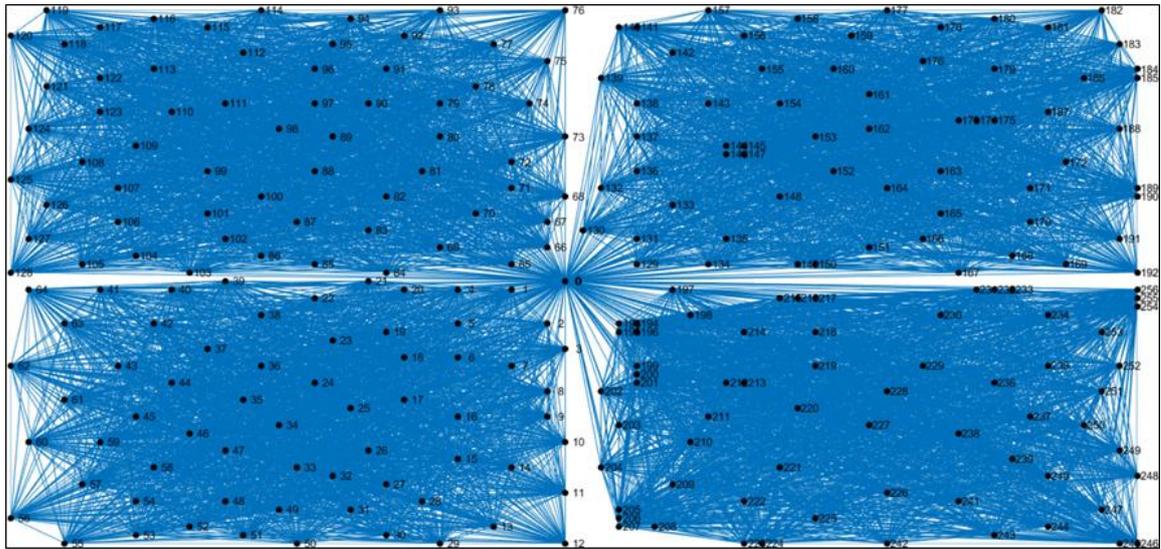
Conforme descrito por Mendes (2016), na resolução de alguns problemas de otimização, especialmente os de larga escala, a geração da população inicial de forma aleatória do Algoritmo Genético pode se tornar inviável, extremamente custosa e/ou ineficiente, sendo necessário substituir o processo randômico por um mecanismo de geração de soluções mais apropriado, capaz de convergirem mais rapidamente e com mais precisão na direção da solução ótima.

Com intuito de melhorar os resultados alcançados e visando potencializar a convergência da solução final encontrada pelo AG, a proposta desse trabalho é utilizar um algoritmo híbrido de otimização (*Hybrid Optimization Algorithm* - HOA) constituído por meio da combinação dos Algoritmos Genético e *Space-Filling Curves*. Sendo essa combinação realizada da seguinte forma:

- a. A área de busca é dividida em quatro partições (*clusters*), conforme a formação da curva de Hilbert de nível 1.
- b. Após a divisão, o grafo da área de busca para a geração das rotas é reduzido em 1/4, ficando os pontos de visitas agrupados em suas respectivas regiões, ligadas apenas pelo ponto que representa o depósito, conforme apresentado na Figura 37. Um ponto importante nessa divisão é que a execução de cada partição é realizada separadamente.
- c. Em seguida, são gerados quatro mapeamentos fractais, um para cada partição, em que a sequência do mapeamento é gerada a partir do ponto central (depósito). Um detalhe importante nesta etapa é que a criação do fractal é realizada no sentido horário, isto é, em cada quadrante o fractal sofre uma rotação 90 graus. A Figura 38 mostra essa proposta.
- d. Na sequência, os pontos são enquadrados nas células da matriz gerada pelo algoritmo fractal, de forma que cada ponto fique em uma célula. A mesma ordem de detalhamento do fractal é aplicada em todos os quadrantes.
- e. Após o posicionamento dos pontos nas células, é gerada uma ordem de mapeamento dos pontos de cada quadrante, conforme a sequência em que as curvas de preenchimento visitam as células da matriz.

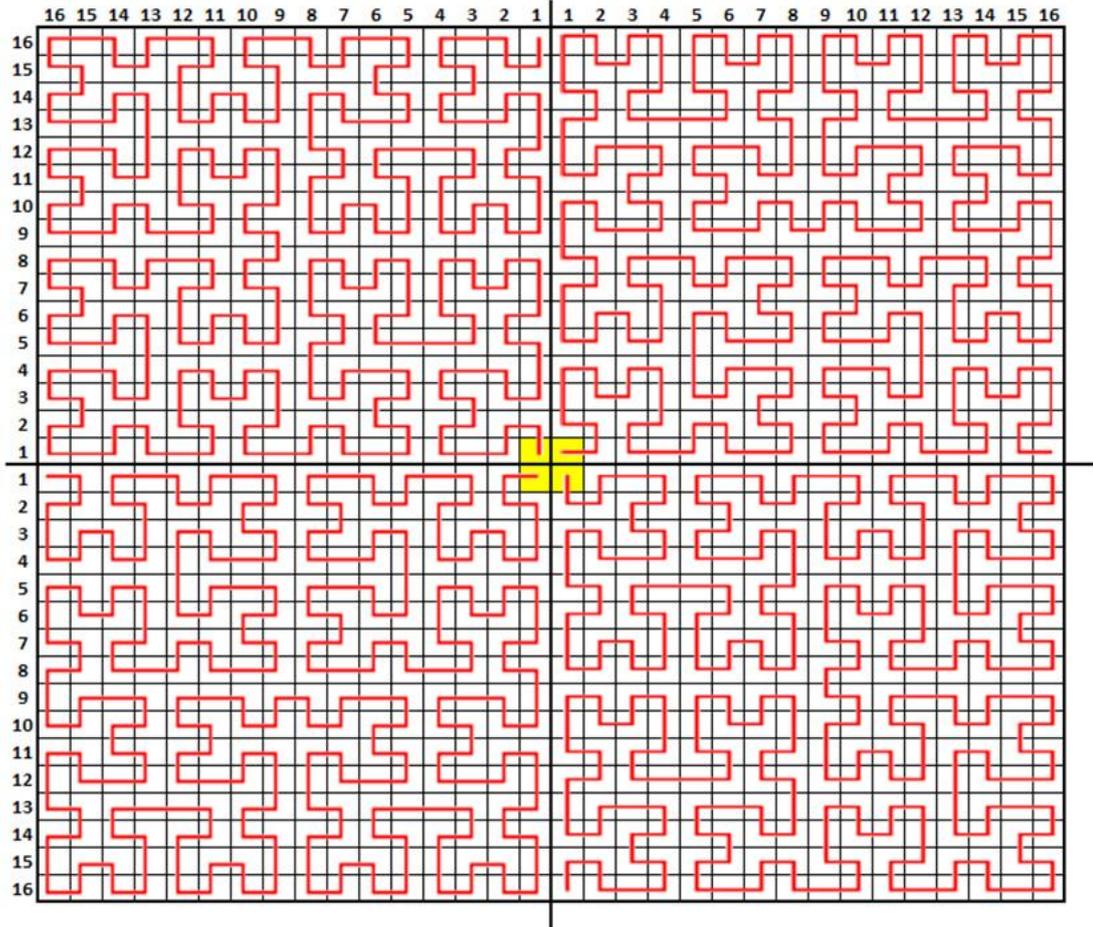
- f. No AG, por definição, a população inicial é gerada aleatoriamente, no entanto, no algoritmo híbrido proposto o primeiro indivíduo da geração inicial é criado pela sequência do mapeamento fractal, que já passa para o AG um superindivíduo formado por genes contendo pontos aproximados, constituindo uma rota sem cruzamento de traços é com certo nível de otimização.
- g. Na criação do superindivíduo, obrigatoriamente o primeiro e o último gene são representados pelo depósito e os demais genes são formado pela sequência de pontos obtidos pelo mapeamento fractal, que sempre inicia do depósito central. A primeira sequência fractal cria um indivíduo da primeira geração do AG para a elaboração da primeira rota da partição em execução. Já a segunda sequência fractal é usada para a criação do indivíduo da primeira geração do AG para a elaboração da segunda rota e assim sucessivamente até que todas as rotas sejam criadas para a partição. Logo, para a elaboração de cada rota da partição, um indivíduo é criado utilizando a sequência do mapeamento fractal.
- h. Como no AG os indivíduos mais aptos são selecionados para formar a novas gerações e também possuem maior probabilidade de serem qualificados para a realização de cruzamento, espera-se que o indivíduo gerado pela sequência fractal prospere por gerações futuras e que seus genes sejam transmitidos para os demais indivíduos da população, contribuindo positivamente para a formação de um indivíduo contendo uma rota mais eficiente, aproximada da solução ótima.
- i. Ao gerar todas as rotas de uma determinada partição, é comum que alguns pontos de vista fiquem de fora devido à restrição do número máximo e mínimo de pontos que devem compor uma rota. Com isso, foi adicionado ao algoritmo uma inteligência para verificar os pontos excedentes existentes na partição atual e os disponibiliza para a próxima. Ao gerar as rotas do próximo quadrante, esses pontos excedentes se tornam factíveis e são incluídos no grafo do espaço de busca da partição em execução, fazendo com que pontos excedentes do quadrante anterior sejam incluídos nas rotas do quadrante em execução.

Figura 37 - Representação do grafo após a divisão da área de busca pela Curva de Hilbert de ordem 1.



Fonte: Próprio autor.

Figura 38 - Mapeamento fractal de cada agrupamento da área de busca.



Fonte: Próprio autor.

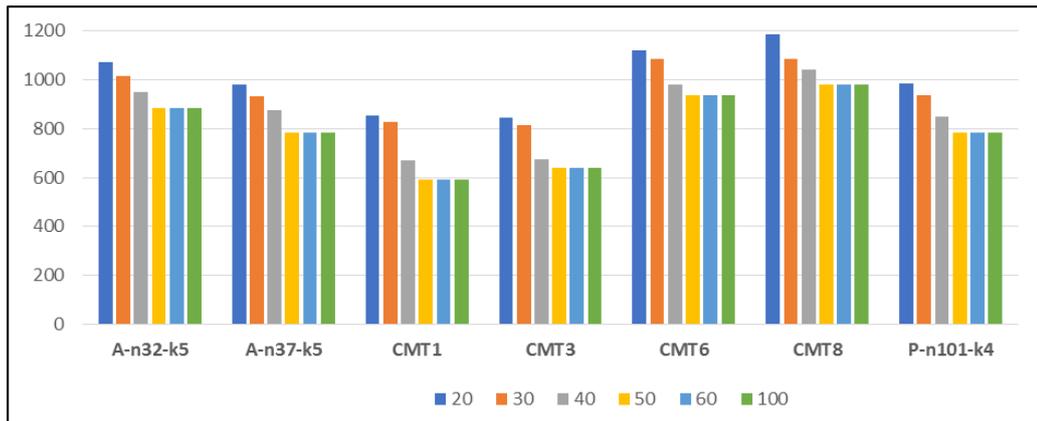
Já no AG, utilizado em combinação com o algoritmo fractal, foi configurado com os seguintes valores:

- Uma população formada por 50 indivíduos em cada geração;
- A população inicial formada por 1 indivíduo gerado pelo algoritmo fractal e os outros 49 formados aleatoriamente por pontos contidos somente na partição da execução;
- Em cada geração, são selecionados pelo método de Elitismo 10 indivíduos de menor custo para participar na nova geração e para realizar cruzamento;
- Também são selecionados pelo método de Roleta outros 15 indivíduos, que também participarão na nova geração e irão realizar cruzamento, seguindo as equações de probabilidades apresentadas na seção 3.1.3.3;
- É realizado o cruzamento dos 25 indivíduos selecionados anteriormente, gerando 25 novos indivíduos, que farão parte da nova geração;
- O cruzamento é realizado pelo método de *Random Crossover*, conforme detalhado na seção 5.1.1;
- Após o cruzamento, cada indivíduo possui uma taxa de 15% de probabilidade de sofrer uma mutação, sendo o *Exchange Mutation* o método utilizado neste ponto, conforme detalhado na seção 5.1.2;
- O ponto de parada do algoritmo é realizado na iteração de número 60 do algoritmo.

Os valores dos parâmetros utilizados no algoritmo híbrido foram balizados por meio de sucessivos experimentos realizados. Os testes foram realizados nos mesmos 10 problemas existentes na base de *benchmark* e após execuções consecutivas para cada valor de parâmetro, as médias dos custos das rotas geradas de 7 instâncias foram representadas na sequência de gráficos mostrados a seguir. O gráfico da Figura 39 apresenta a calibração do algoritmo para determinar o valor para o parâmetro de número de indivíduos da população, para cada geração. Nele é

possível perceber que o melhor valor encontrado pelo algoritmo foi de uma população formada por 50 indivíduos.

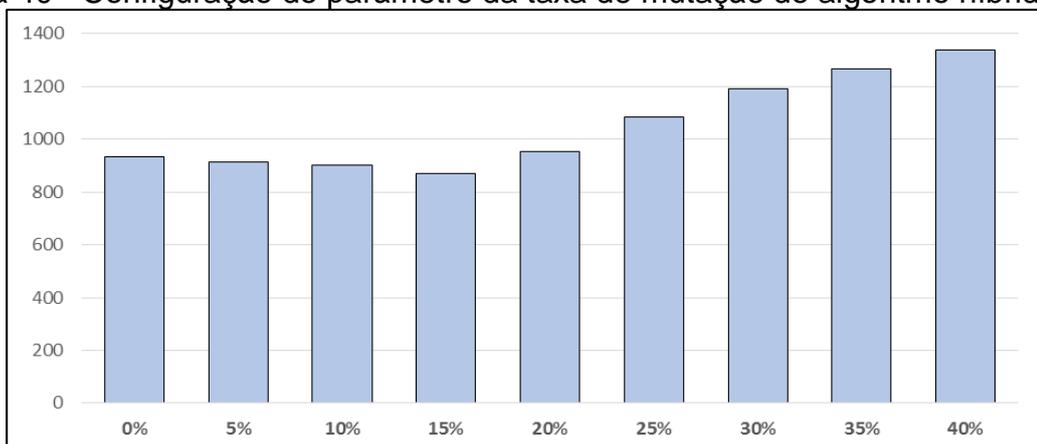
Figura 39 - Calibração do parâmetro do número de indivíduos da população do algoritmo híbrido.



Fonte: Próprio autor.

O gráfico da Figura 40 apresenta os resultados dos sucessivos experimentos para a calibração do parâmetro da taxa de probabilidade de mutação. Nesse gráfico é possível perceber que o melhor valor de probabilidade é de 15%, que significa que após a realização do cruzamento, cada indivíduo na nova geração terá 15% chance de sofrer mutação.

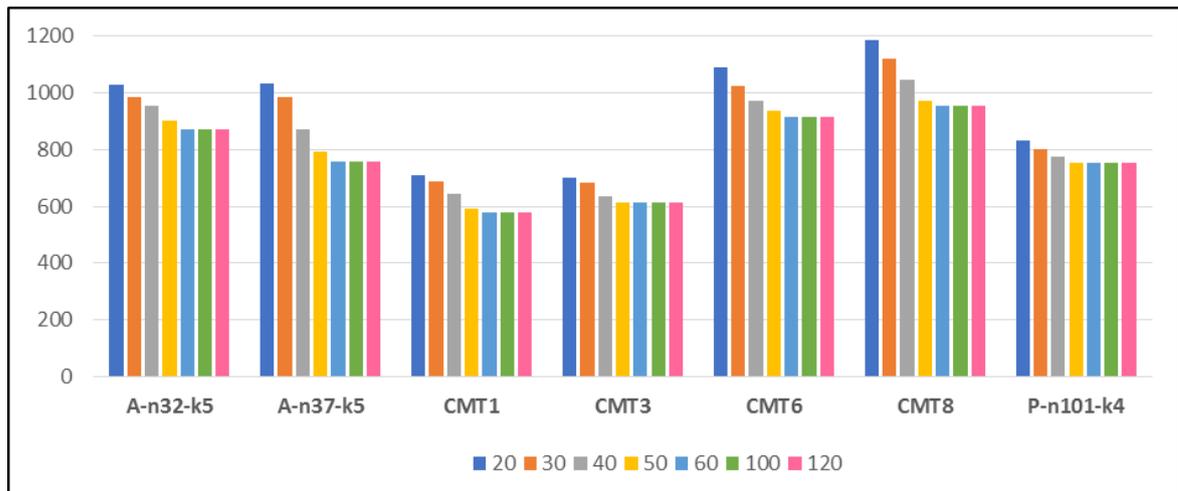
Figura 40 - Configuração do parâmetro da taxa de mutação do algoritmo híbrido.



Fonte: Próprio autor.

Já o gráfico da Figura 41 apresenta o valor de calibração do critério de parada do algoritmo, tendo encontrado a melhor solução na iteração de número 60.

Figura 41 - Calibração do critério de parada do algoritmo híbrido.



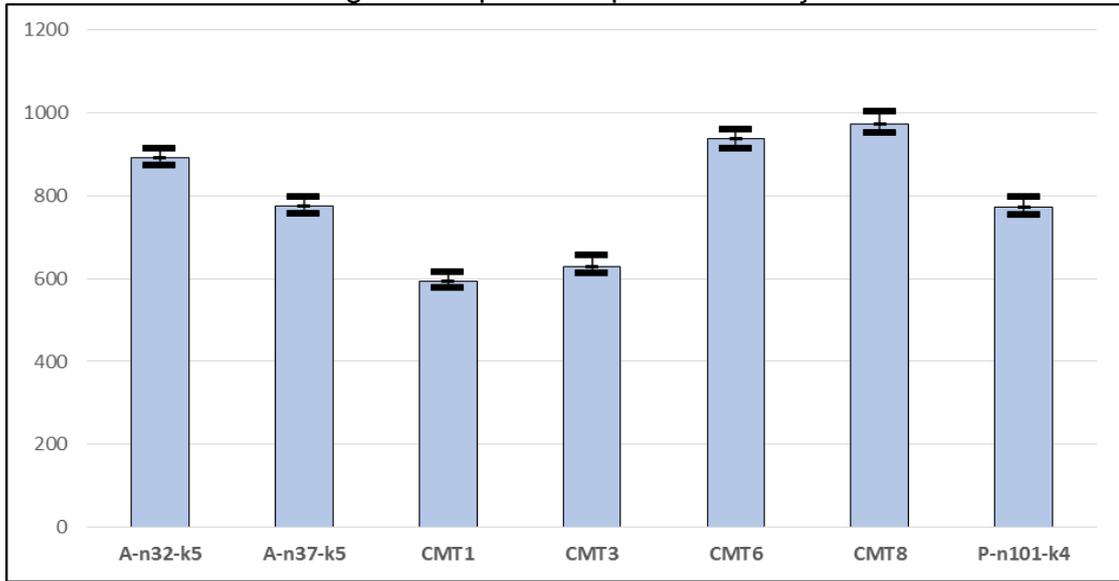
Fonte: Próprio autor.

5.3 RESULTADOS E COMPARAÇÕES

Após a etapa de calibração dos parâmetros do algoritmo, ele foi novamente executado 10 vezes e os valores alcançados são apresentados no gráfico da Figura 42. É possível verificar no gráfico que os valores máximos e mínimos de cada experimento ficaram bem próximos aos valores da média, demonstrando a baixa variação dos resultados do algoritmo.

A Tabela 11 apresenta uma comparação entre os resultados de todas as abordagens testadas ao longo deste trabalho. A tabela demonstra um alcance médio de 13,3% do resultado em comparação com a solução ótima, que comprova a viabilidade do algoritmo híbrido proposto.

Figura 42 - Resultado do algoritmo após a etapa de calibração.



Fonte: Próprio autor.

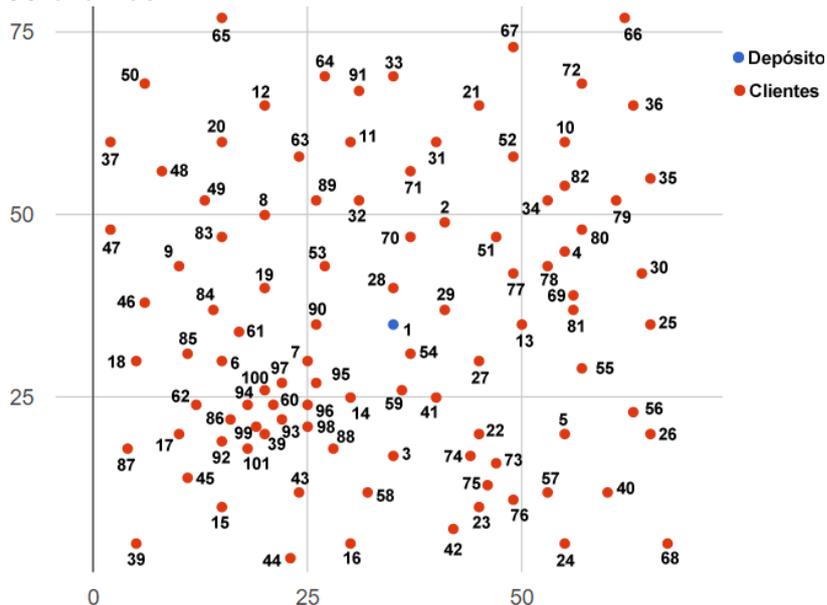
Tabela 11 - Comparação entre todas as abordagens experimentadas.

Instância	Nº de Clientes + 1 depósito	Nº de Veículos	Solução Ótima			Algoritmo Híbrido (SEC + GA)			Algoritmo Genético com melhorias			Algoritmo Genético impl. padrão			
			Custo	Diferença entre os Custos	Tempo de Exec. (s)	Custo	Diferença entre os Custos	Tempo de Exec. (s)	Custo	Diferença entre os Custos	Tempo de Exec. (s)	Custo	Diferença entre os Custos	Tempo de Exec. (s)	
A-n32-k5	32	5	784	892	108	13,8%	452	937	153	19,5%	316	1003	219	27,9%	260
A-n33-k5	33	5	661	752	91	13,8%	453	789	128	19,4%	320	844	183	27,7%	263
A-n33-k6	33	6	742	844	102	13,7%	462	887	145	19,5%	332	949	207	27,9%	266
A-n34-k5	34	5	778	889	111	14,3%	477	929	151	19,4%	340	993	215	27,6%	266
A-n36-k5	36	5	799	918	119	14,9%	478	954	155	19,4%	352	1020	221	27,7%	272
A-n37-k5	37	5	669	766	97	14,5%	479	802	133	19,9%	359	857	188	28,1%	275
A-n37-k6	37	6	949	1089	140	14,8%	483	1118	169	17,8%	361	1212	263	27,7%	281
A-n38-k5	38	5	730	837	107	14,7%	483	865	135	18,5%	360	933	203	27,8%	281
A-n39-k5	39	5	822	942	120	14,6%	485	980	158	19,2%	360	1052	230	28,0%	284
A-n39-k6	39	6	831	952	121	14,6%	484	989	158	19,0%	364	1059	228	27,4%	287
A-n44-k6	44	6	937	1075	138	14,7%	487	1109	172	18,4%	376	1195	258	27,5%	292
A-n45-k6	45	6	944	1082	138	14,6%	487	1120	176	18,6%	376	1207	263	27,9%	292
A-n45-k7	45	7	1146	1312	166	14,5%	490	1355	209	18,2%	378	1464	318	27,7%	293
A-n46-k7	46	7	914	1044	130	14,2%	494	1085	171	18,7%	378	1167	253	27,7%	293
A-n48-k7	48	7	1073	1222	149	13,9%	496	1272	199	18,5%	380	1369	296	27,6%	294
A-n53-k7	53	7	1010	1148	138	13,7%	501	1193	183	18,1%	392	1291	281	27,8%	302
A-n54-k7	54	7	1167	1328	161	13,8%	502	1378	211	18,1%	393	1491	324	27,8%	303
A-n55-k9	55	9	1073	1218	145	13,5%	506	1264	191	17,8%	397	1372	299	27,9%	305
A-n60-k9	60	9	1354	1540	186	13,7%	510	1589	235	17,4%	413	1735	381	28,1%	308
A-n61-k9	61	9	1034	1172	138	13,3%	512	1217	183	17,7%	413	1322	288	27,9%	308
A-n62-k8	62	8	1288	1460	172	13,4%	511	1515	227	17,6%	406	1649	361	28,0%	311
A-n63-k9	63	9	1616	1835	219	13,6%	515	1898	282	17,5%	407	2066	450	27,8%	316
A-n63-k10	63	10	1314	1492	178	13,5%	517	1546	232	17,7%	414	1676	362	27,5%	317
A-n64-k9	64	9	1401	1589	188	13,4%	516	1647	246	17,6%	411	1788	387	27,6%	316
A-n65-k9	65	9	1174	1327	153	13,0%	517	1383	209	17,8%	419	1497	323	27,5%	318
A-n69-k9	69	9	1159	1311	152	13,1%	520	1363	204	17,6%	419	1478	319	27,5%	322
A-n80-k10	80	10	1763	1990	227	12,9%	548	2071	308	17,5%	502	2250	487	27,6%	376
CMT1	50	5	524	595	71	13,5%	511	615	91	17,4%	442	667	143	27,3%	318
CMT2	75	10	835	938	103	12,3%	544	982	147	17,6%	472	1067	232	27,8%	364
CMT3	100	8	555	628	73	13,2%	553	653	98	17,7%	527	712	157	28,3%	375
CMT4	150	12	1028	1157	129	12,5%	702	1209	181	17,6%	539	1315	287	27,9%	402
CMT5	199	16	1291	1443	152	11,8%	732	1518	227	17,6%	552	1642	351	27,2%	409
CMT6	50	6	826	937	111	13,4%	527	978	152	18,4%	378	1053	227	27,5%	286
CMT7	75	11	909	1003	94	10,3%	548	1069	160	17,6%	791	1158	249	27,4%	382
CMT8	100	9	865	974	109	12,6%	556	1014	149	17,2%	447	1109	244	28,2%	353
P-n101-k4	101	4	681	774	93	13,7%	545	803	122	17,9%	441	867	186	27,3%	354
X-n115-k10	115	10	12747	14128	1381	10,8%	590	15102	2355	18,5%	496	16125	3378	26,5%	383
X-n139-k10	139	10	13590	14952	1342	9,9%	674	16017	2427	17,9%	514	17292	3702	27,2%	392
X-n256-k16	256	16	18839	20728	1889	10,0%	769	22022	3183	16,9%	567	24156	5317	28,2%	413
					Média:	13,3%			Média:	18,2%			Média:	27,7%	

Fonte: Próprio autor.

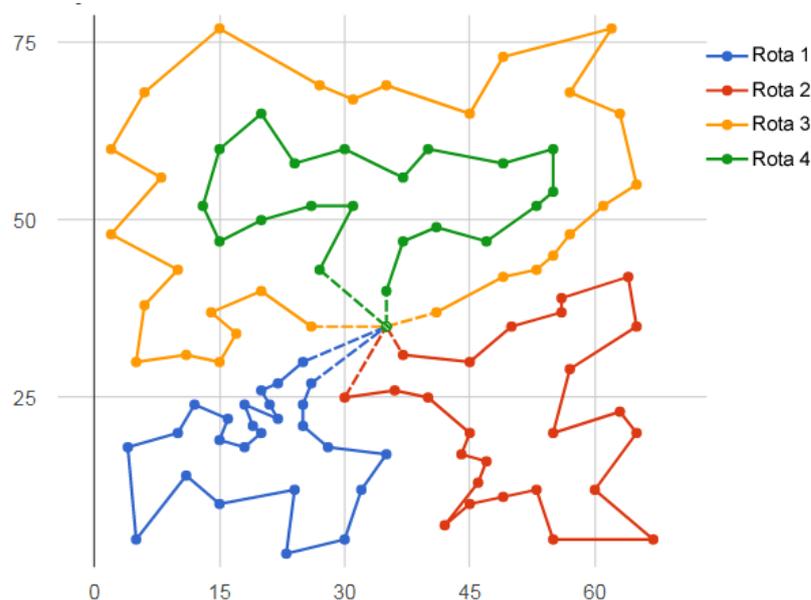
A sequência de imagens da Figura 43 a 45 apresentam um resultado da comparação realizada entre o algoritmo proposto e sua respectiva solução ótima obtida pelo trabalho de (UCHOA et al., 2017).

Figura 43 - Instância P-n101-k4 da base de *benchmark*, espaço de busca contendo 4 veículos e 100 clientes.



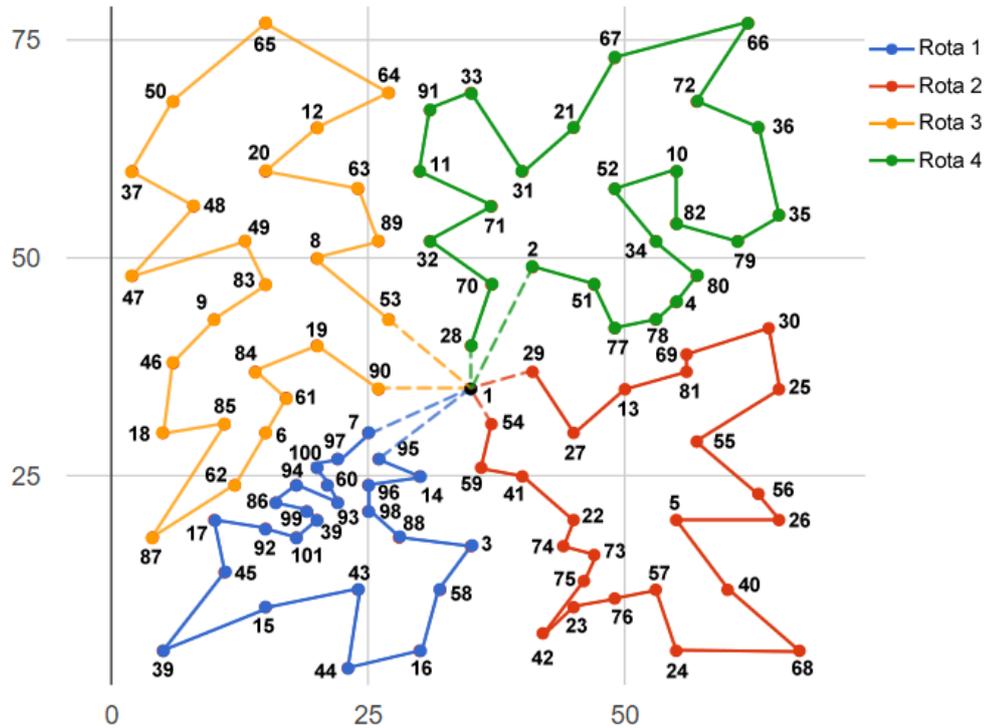
Fonte: Uchoa et al. (2017).

Figura 44 - Solução ótima contendo custo de 681, apresentada por Uchoa et al. (2017).



Fonte: Uchoa et al. (2017).

Figura 45 - Resultado do algoritmo híbrido contendo custo de 754, uma diferença de 10,7% da solução ótima.



Fonte: Próprio autor.

Após utilizar da base de *benchmark* publicada por Uchoa et al. (2017) para a realização de testes, calibração, comparação e validação do algoritmo, uma nova base foi testada, com intuito de verificar os resultados alcançados pelo algoritmo híbrido proposto em relação a outros algoritmos híbridos existentes na literatura.

No trabalho publicado por Necula, Breaban e Raschip (2015a), os autores disponibilizam uma nova base de teste e realizaram uma avaliação de 5 variações do Algoritmo de Colônia de Formigas (*Ant Colony Optimization - ACO*), aplicado à resolução do *Single-Depot Multiple Traveling Salesman Problem*. As variações do ACO utilizadas pelos autores foram:

- Problema de decomposição com k-Means seguida por ACS para TSP (kM-ACS);
- ACS com atualização de feromônio de solução global (g-ACS);
- ACS com atualização de feromônio no subtour (s-ACS);

- ACS com atualização de feromônio na solução global e tours delimitados (gb-ACS);
- ACS com atualização de feromônio no subtour e tours delimitados (sb-ACS).

Com o objetivo de verificar o desempenho do algoritmo híbrido proposto em relação aos resultados alcançados pelos trabalhos de Necula, Breaban e Raschip (2015a, 2015b), o mesmo foi executado na mesma base de teste criada pelos autores, especificamente nas instâncias:

- eil51: formada por 1 depósito e 50 cidades;
- eil76: formada por 1 depósito e 75 cidades;
- rat99: formada por 1 depósito e 98 cidades.

Ainda, cada instância foi testada com 2, 3, 5 e 7 veículos para a realização das rotas, tendo como único objeto minimizar o custo total de visita a todas as cidades.

Em seu trabalho, os autores Necula, Breaban e Raschip (2015a) apresentam o tempo de execução no CPLEX para a instância eil51 com 7 veículos e para a instância rat99 com 3, 5 e 7 veículos. Essas informações são apresentadas na Tabela 12, que também demonstra uma comparação do tempo de execução entre o CPLEX e o algoritmo híbrido proposto por este trabalho na resolução desses problemas. Já a Tabela 13 apresenta uma comparação entre os resultados alcançados pela combinação dos Algoritmos Genético e *Space-Filling Curves* em relação às cinco variações do ACO utilizadas pelos autores em Necula, Breaban e Raschip (2015a).

Tabela 12 - Comparação do tempo de execução entre o CPLEX e o Algoritmo Híbrido.

Instância	M	CPLEX		Algoritmo Híbrido (SFC + GA)	
		Tempo	Custo	Tempo	Custo
eil51	7	120 hrs	605,21	642 seg	631,59
rat99	3	96 hrs	1519,49	560 seg	1.632,61
	5	168 hrs	1855,83	684 seg	2.055,74
	7	216 hrs	2291,82	701 seg	2.703,38

Fonte: Próprio autor.

Tabela 13 - Comparação dos resultados alcançados por cada algoritmo.

Instância	m	Solução Ótima	kM-ACS		g-ACS		s-ACS		gb-ACS		sb-ACS		SFC + GA	
			Custo	Diferença	Custo	Diferença	Custo	Diferença	Custo	Diferença	Custo	Diferença	Custo	Diferença
eil51	2	442,32	454,30	2,7%	452,66	2,3%	454,96	2,9%	452,22	2,2%	453,81	2,6%	451,60	2,1%
	3	464,11	500,00	7,7%	485,73	4,7%	489,64	5,5%	479,51	3,3%	483,39	4,2%	482,14	3,9%
	5	529,70	563,58	6,4%	582,36	9,9%	590,63	11,5%	585,76	10,6%	598,61	13,0%	570,73	7,7%
	7	605,21	634,47	4,8%	674,78	11,5%	680,38	12,4%	688,26	13,7%	699,47	15,6%	642,54	6,2%
			Média:	5,4%	Média:	7,1%	Média:	8,1%	Média:	7,5%	Média:	8,8%	Média:	5,0%
eil76	2	558,59	594,21	6,4%	580,77	4,0%	583,41	4,4%	579,68	3,8%	578,96	3,6%	572,64	2,5%
	3	579,30	642,89	11,0%	622,91	7,5%	630,67	8,9%	613,76	5,9%	619,19	6,9%	592,83	2,3%
	5	680,67	740,35	8,8%	747,49	9,8%	760,05	11,7%	734,61	7,9%	744,94	9,4%	707,67	4,0%
	7	759,90	820,35	8,0%	873,65	15,0%	883,63	16,3%	894,70	17,7%	911,06	19,9%	818,57	7,7%
			Média:	8,5%	Média:	9,1%	Média:	10,3%	Média:	8,8%	Média:	10,0%	Média:	4,1%
rat99	2	1.350,73	1.485,56	10,0%	1.398,01	3,5%	1.398,01	3,5%	1.391,89	3,0%	1.382,05	2,3%	1.407,40	4,2%
	3	1.519,49	1.672,11	10,0%	1.691,56	11,3%	1.707,20	12,4%	1.661,04	9,3%	1.651,68	8,7%	1.643,74	8,2%
	5	1.855,83	1.996,04	7,6%	2.260,74	21,8%	2.297,05	23,8%	2.286,73	23,2%	2.337,94	26,0%	2.074,27	11,8%
	7	2.291,82	2.361,55	3,0%	2.859,98	24,8%	2.878,97	25,6%	3.004,37	31,1%	2.984,42	30,2%	2.727,41	19,0%
			Média:	7,7%	Média:	15,4%	Média:	16,3%	Média:	16,7%	Média:	16,8%	Média:	10,8%

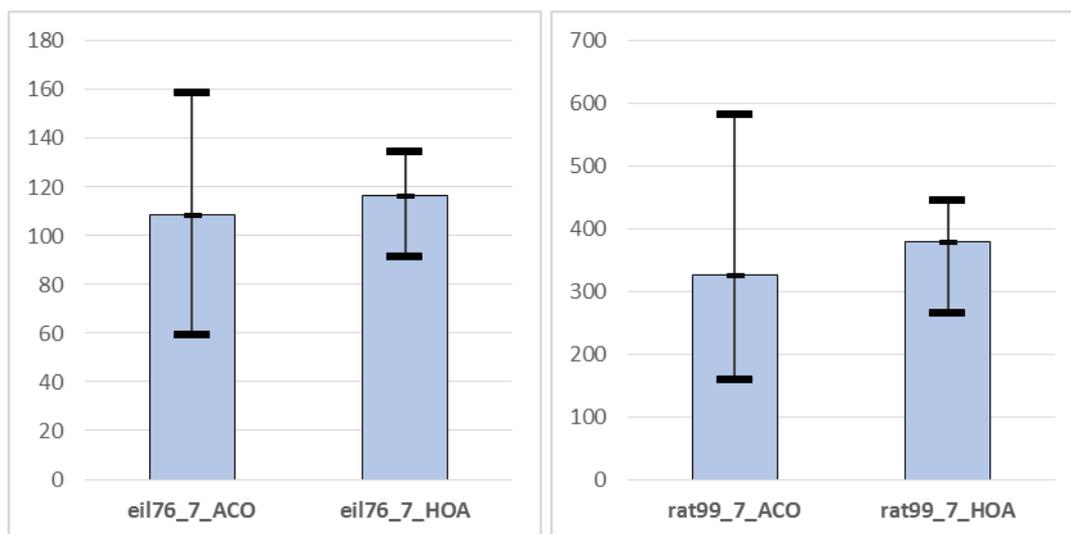
Fonte: Próprio autor.

Como é possível perceber ao analisar a tabela anterior, o algoritmo proposto alcançou um resultado superior na resolução de quase todos os problemas testados, ficando, por exemplo, a 2,1% da solução ótima na resolução do problema eli51-M2 e 2,3% na resolução do problema eli76-M3.

Os gráficos apresentados na Figura 46 demonstram uma comparação da variação entre a rota de maior custo, a rota de menor custo e o custo médio de uma solução encontrada pelo algoritmo dos autores Necula, Breaban e Raschip (2015a) e o algoritmo híbrido proposto. Ao analisar os gráficos é possível observar que no algoritmo proposto a variação foi significativamente menor, devido a capacidade de o algoritmo gerar rotas mais balanceadas, com uma diferença pequena entre os números de pontos de vistas de cada rota. Essa restrição foi adicionada ao algoritmo

para possibilitar a sua utilização em um sistema comercial, equilibrando a jornada de trabalho os empregados que irão realizar os percursos e evitando sobrecarregar algum empregado em específico no atendimento aos clientes.

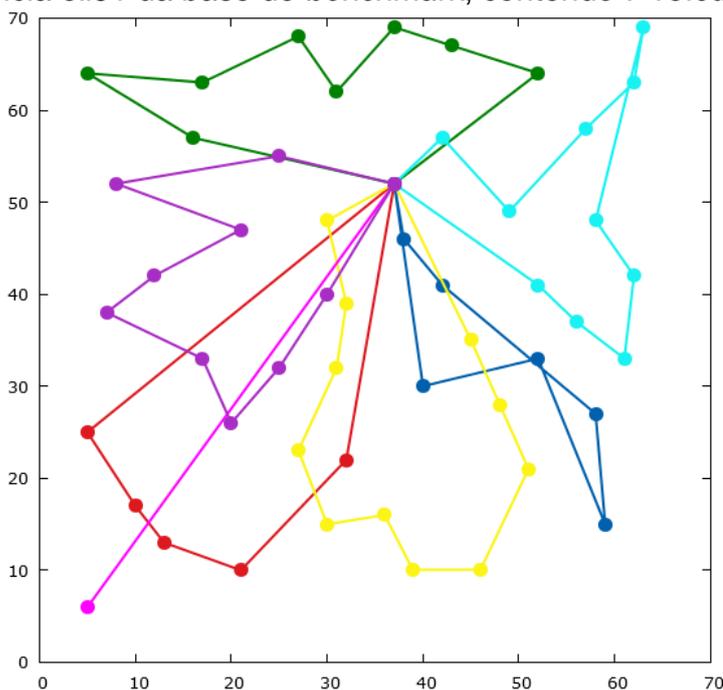
Figura 46 - Comparação entre a variação dos custos das rotas.



Fonte: Próprio autor.

A sequência de imagens das Figuras 47 até 50 apresentam algumas comparações entre os resultados do algoritmo dos autores Necula, Breaban e Raschip (2015a) e o algoritmo híbrido proposto.

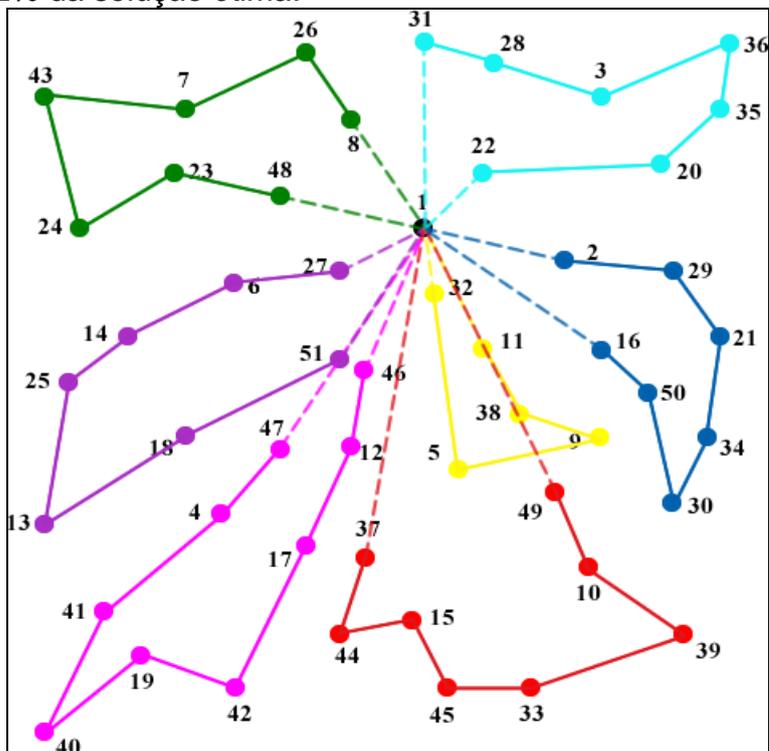
Figura 47 - Instância eil51 da base de *benchmark*, contendo 7 veículos e 50 clientes.



Resolução proposta por Necula, Breaban e Raschip (2015a) com custo final de 762,83.

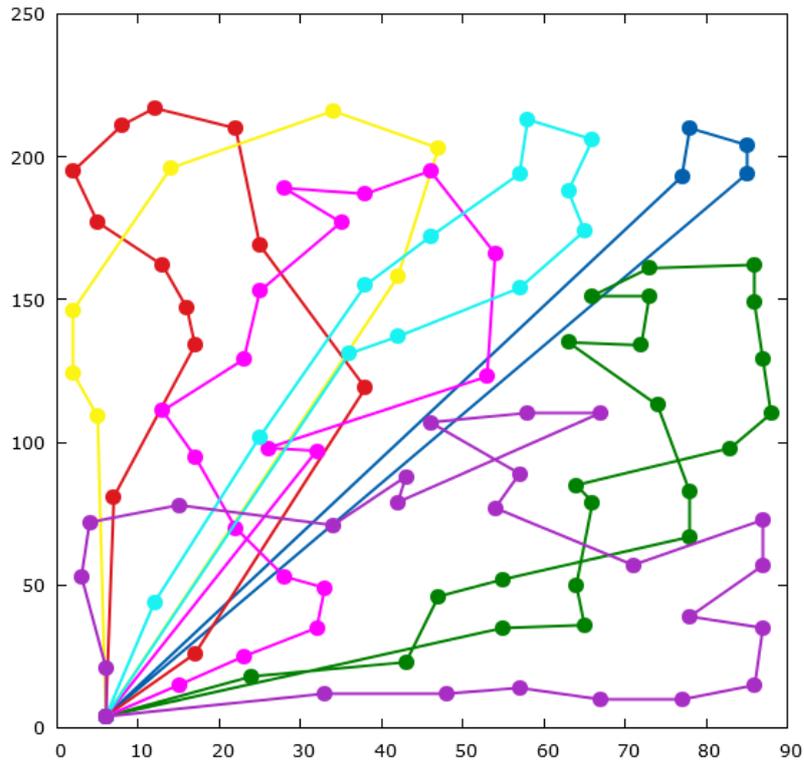
Fonte: Necula, Breaban e Raschip (2015a).

Figura 48 - Resultado do algoritmo híbrido proposto com custo final de 631,59, uma diferença de 6,2% da solução ótima.



Fonte: Próprio autor.

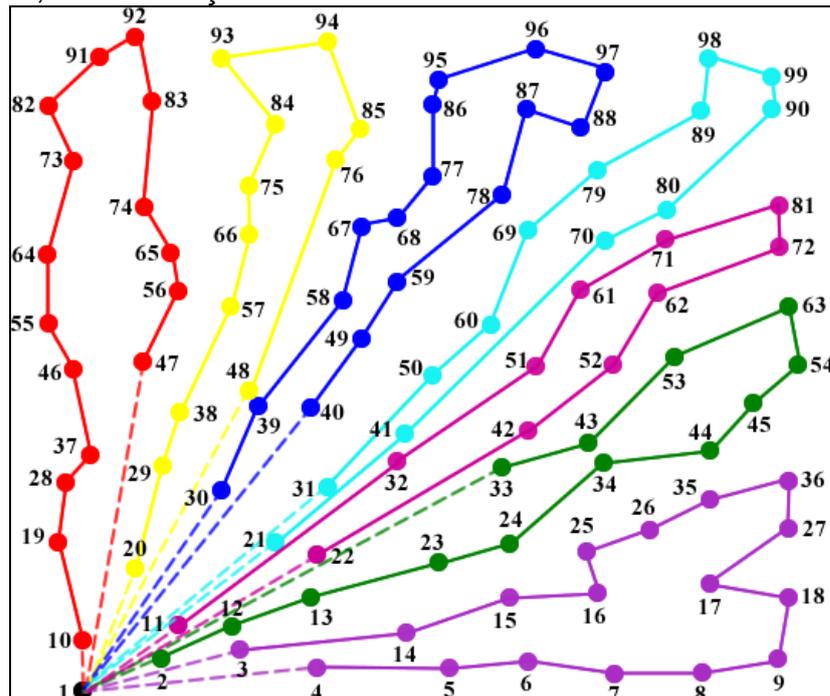
Figura 49 - Instância rat99 da base de *benchmark*, contendo 7 veículos e 98 clientes.



Resolução proposta por Necula, Breaban e Raschip (2015a) com custo final de 3074,30.

Fonte: Necula, Breaban e Raschip (2015a).

Figura 50 - Resultado do algoritmo híbrido proposto com custo final de 2703,38, uma diferença de 18,0% da solução ótima.



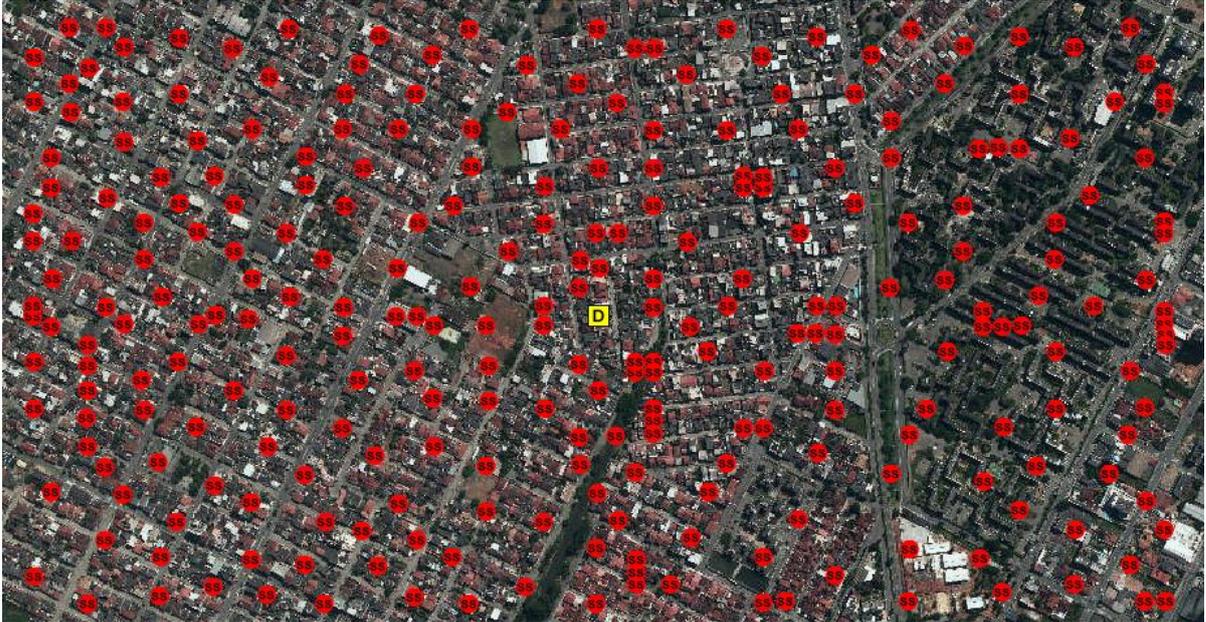
Fonte: Próprio autor.

Como é possível perceber, ao analisar a sequência de imagens apresentadas anteriormente, todas as rotas foram geradas sem a existência de cruzamento de traços e o algoritmo proposto alcançou um bom nível de otimização, quando comparado com a solução ótima. Além disso, as rotas também foram geradas com quantidades balanceadas de pontos, o que na prática significa que os veículos que realizarão os trajetos não ficaram sobrecarregados, demonstrando a viabilidade de utilização do algoritmo em uma aplicação comercial.

Após comprovada a viabilidade o algoritmo proposto, por meio dos diversos testes e comparações realizadas, o mesmo foi testado diretamente nas informações de solicitações de serviços existentes na base de dados do sistema GIS Corporativo da CESAN. A Figura 51 mostra a distribuição de 256 pontos de SS georreferenciados no município de Vila Velha, representados pela cor vermelha, e o depósito desse município, representado pela cor amarela.

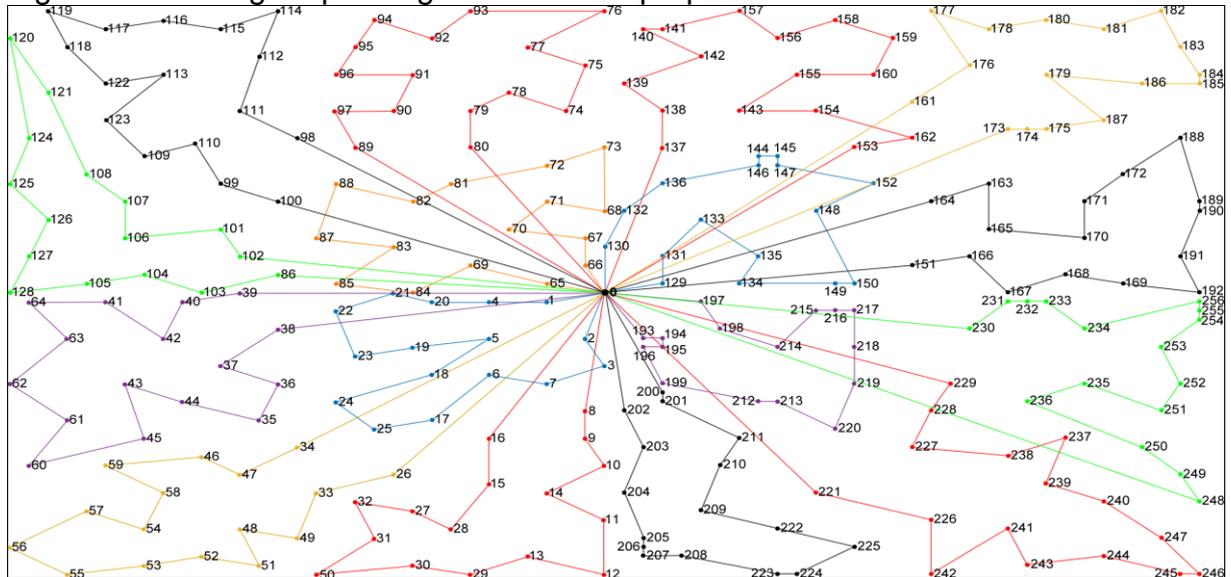
Já a Figura 52 apresenta as rotas geradas pelo algoritmo baseando-se nas restrições apresentadas na seção 4.1 deste trabalho, que determinam que ao longo de um dia um empregado percorre uma rota contendo, no máximo, 16 e, no mínimo, 14 pontos de inspeção. Com isso, o algoritmo gerou um conjunto de 16 rotas equilibradas, em que cada uma contém 16 pontos a serem visitados, alcançando um custo total das rotas de 17.883 metros.

Figura 51 - Mapa do sistema GIS contendo 256 pontos de Solicitação de Serviços georreferenciados.



Fonte: Próprio autor.

Figura 52 - Rota gera pelo algoritmo híbrido proposto de custo final 17.883 metros.

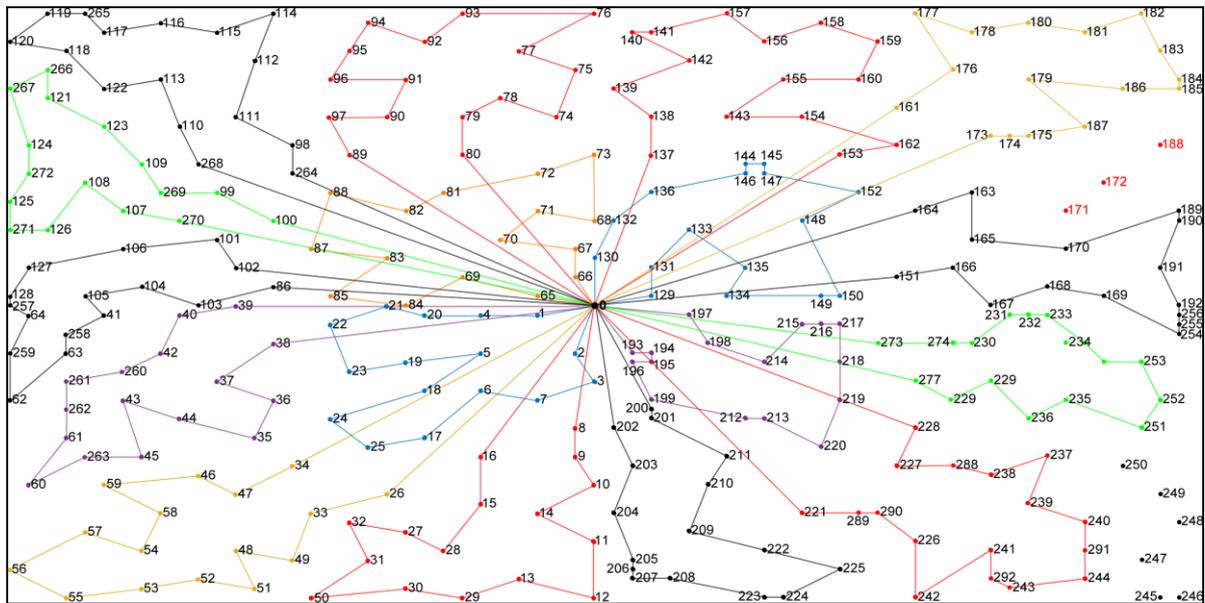


Fonte: Próprio autor.

Em seguida, com intuito de simular a dinâmica do planejamento de rotas da CESAN, um total de 25 novos pontos foram adicionados ao problema e 3 pontos foram removidos (171, 172, 188). Em seguida o programa foi reprocessado sem levar em consideração a restrição que limita em 16 o número máximo rotas. Com isso, o

algoritmo gerou 17 rotas, contendo 16 pontos de visita cada, deixando 6 pontos de fora, que farão parte das rotas a serem geradas no dia seguinte. A Figura 53 apresenta o conjunto de rotas geradas pela a aplicação, com custo total de 19.477 metros.

Figura 53 - Rota gera pelo algoritmo híbrido proposto com custo final de 19.477 metros.



Fonte: Próprio autor.

Atualmente, o algoritmo híbrido proposto, utilizado na criação do módulo de geração automática de rotas no sistema GIS Corporativo, foi implementado utilizando diretamente as APIs de geoprocessamento do ArcGIS, em conjunto com a linguagem .Net C# e Adobe Flex.

6 CONCLUSÃO

Esse trabalho propôs e implementou um Algoritmo Híbrido constituído da combinação dos algoritmos Genético e *Space-Filling Curves*, em que a curva de Hilbert foi utilizada para a criação de parte dos indivíduos da primeira geração do Algoritmo Genético. Conforme demonstrado ao longo desse estudo, essa abordagem permitiu uma convergência mais rápida do algoritmo e potencializou a aplicação do Algoritmo Genético, contribuindo positivamente para um resultado da solução final de até 10,5% próxima ao da solução ótima quando comparado ao trabalho desenvolvido por Uchoa et al. (2017).

Além disso, também houve uma preocupação na elaboração do algoritmo proposto para que o mesmo gerasse soluções balanceadas, em que todas as rotas tivessem um número de pontos de visitas iguais ou com uma quantidade aproximada, balanceando a jornada de trabalho os empregados que irão realizar os percursos, alinhado à restrição do número de veículos existentes na empresa para a realização dos roteiros e também da jornada de trabalho de 8 horas diárias dos empregados, que devem atender no máximo 16 pontos de visitas ao longo de um dia.

Os diversos experimentos realizados nas duas bases distintas de *benchmark* comprovaram que a solução proposta apresenta bons resultados frente à métodos disponíveis na literatura na resolução do problema de *Single Depot VRP*, como a comparação realizada com o trabalho de Necula, Breaban e Raschip (2015a), que utilizou 5 variações do Algoritmo de Colônia de Formigas. Nessa comparação foi demonstrado que o algoritmo híbrido proposto pelo presente trabalho foi superior na resolução de quase todos os problemas contidos nas instâncias *eil51*, *eil76* e *rat99* da base de dados criada pelos próprios autores do trabalho. Além disso, o algoritmo alcançou resultados muito próximos da solução ótima, chegando a ficar, por exemplo, a 0,1% da solução ótima na resolução do problema *eli51-M2* e 0,9% na resolução do problema *eli76-M3*.

Como a motivação inicial desse trabalho foi direcionada à resolução do problema de roteamento de veículos existente na Companhia Espírito Santense de Saneamento,

e também alinhado à proposta do Mestrado Profissional, o resultado final deste trabalho culminou na criação de um novo módulo de geração automática de rotas de menor custo para informações georreferenciadas, existentes no sistema da GIS Corporativo, que encontra-se em fase de homologação pelos usuários da empresa e que em breve irá substituir todo o processo manual de geração de rotas existente atualmente.

6.1 TRABALHOS FUTUROS

Como proposta de continuidade do trabalho, sugere-se a utilização de outra curva de preenchimento de espaço, como a curva de Sierpinski (BARTHOLDI III, 2017b), em substituição das curvas de Hilbert. Um ponto importante na utilização da curva de Sierpinski é que ele também pode ser utilizado para realizar a clusterização da área de busca.

Além disso, também pode ser verificada a possibilidade de combinar o algoritmo fractal com outras heurísticas, por exemplo, o Algoritmo de Colônia de Formiga ou o Enxame de Partículas.

Esse trabalho também pode ser evoluído para tratar a resolução de outros problemas de VRPs, como o *VRP with Time Windows* e o *Pickup-and-Delivery VRP*, além de outros problemas de busca, como rotas de menor custo para robôs e desvios de obstáculos.

REFERÊNCIAS

AGÊNCIA DE REGULAÇÃO DE SERVIÇOS PÚBLICOS DO ESPÍRITO SANTO. **Saneamento básico**. Disponível em: <<https://arsp.es.gov.br/saneamento-basico>>. Acesso em: 20 out. 2017.

BALAS, Egon; CHRISTOFIDES, Nicos. A restricted lagrangean approach to the traveling salesman problem. **Mathematical Programming**, v. 21, n. 1, p. 19-46, 1981.

BADER, Michael. **Space-filling curves**: an introduction with applications in scientific computing. Springer Science & Business Media, 2012.

BARTHOLDI III, John J.; PLATZMAN, Loren K. Spacefilling curves and the planar travelling salesman problem. **Journal of the ACM (JACM)**, v. 36, n. 4, p. 719-737, 1989.

BARTHOLDI III, John J.; PLATZMAN, Loren K. Heuristics based on spacefilling curves for combinatorial problems in Euclidean space. **Management Science**, v. 34, n. 3, p. 291-305, 1988.

BARTHOLDI III, John J. **A routing system based on spacefilling curves**. 2003. Disponível em: <<http://www2.isye.gatech.edu/~jjb/research/mow/mow.pdf>>. Acesso em: 10 out. 2017a.

BARTHOLDI III, J. J. **Some combinatorial applications of spacefilling curves**. Disponível em: <<https://www2.isye.gatech.edu/~jjb/research/mow/mow.html>>. Acesso em: 10 out. 2017b.

BARTHOLDI III, John J.; GOLDSMAN, Paul. Vertex-labeling algorithms for the Hilbert spacefilling curve. **Software: Practice and Experience**, v. 31, n. 5, p. 395-408, 2001a.

BARTHOLDI III, John J.; GOLDSMAN, Paul. Continuous indexing of hierarchical subdivisions of the globe. **International Journal of Geographical Information Science**, v. 15, n. 6, p. 489-522, 2001b.

BARTHOLDI III, John J. et al. A minimal technology routing system for meals on wheels. **Interfaces**, v. 13, n. 3, p. 1-8, 1983.

BEKTAS, Tolga. The multiple traveling salesman problem: an overview of formulations and solution procedures. **Omega**, v. 34, n. 3, p. 209-219, 2006.

BRAEKERS, Kris; RAMAEKERS, Katrien; VAN NIEUWENHUYSE, Inneke. The vehicle routing problem: State of the art classification and review. **Computers & Industrial Engineering**, v. 99, p. 300-313, 2016.

CACERES-CRUZ, Jose et al. Rich vehicle routing problem: Survey. **ACM Computing Surveys (CSUR)**, v. 47, n. 2, p. 32, 2015.

CHRISTOFIDES, Nicos; MINGOZZI, Aristide; TOTH, Paolo. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. **Mathematical programming**, v. 20, n. 1, p. 255-282, 1981.

CLARKE, Geoff; WRIGHT, John W. Scheduling of vehicles from a central depot to a number of delivery points. **Operations research**, v. 12, n. 4, p. 568-581, 1964.

DANTZIG, George B.; RAMSER, John H. The truck dispatching problem. **Management science**, v. 6, n. 1, p. 80-91, 1959.

DENG, Yong; LIU, Yang; ZHOU, Deyun. An improved genetic algorithm with initial population strategy for symmetric TSP. **Mathematical Problems in Engineering**, v. 2015, 2015. Disponível em: <<http://dx.doi.org/10.1155/2015/212794>>.

FITZ, Paulo Roberto. **Geoprocessamento sem complicação**. São Paulo: Oficina de textos, 2008.

GIGERENZER, Gerd; GAISSMAIER, Wolfgang. Heuristic decision making. **Annual review of psychology**, v. 62, p. 451-482, 2011.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 2002.

GOLDBARG, Marco Cesar; LUNA, Henrique Pacca L. **Otimização combinatória e programação linear: modelos e algoritmos**. Rio de Janeiro: Elsevier, 2005.

GOLDBERG, David E. Genetic algorithms in search, optimization, and machine learning. **Reading: Addison-Wesley**, 1989.

GOLDEN, Bruce L.; RAGHAVAN, Subramanian; WASIL, Edward A. (Ed.). **The vehicle routing problem: latest advances and new challenges**. Springer Science & Business Media, 2008. Disponível em: <<http://dx.doi.org/10.1007/978-0-387-77778-8>>.

GORGULHO JÚNIOR, José Hamilton Chaves. **Curvas de preenchimento espacial**. Universidade Federal de Itajubá. Instituto de Engenharia de Produção e Gestão. Disponível em: <<http://www.iem.unifei.edu.br/gorgulho/eme006/eme006.html>>. Acesso em: 10 out. 2017.

HEINEN, Milton Roberto; OSÓRIO, Fernando Santos. Algoritmos genéticos aplicados ao problema de roteamento de veículos. **HÍFEN**, v. 30, n. 58, 2006.

HILBERT, David. Mathematische probleme. Nachrichten von der Königlichen Gesellschaft der Wissenschaften zu Göttingen, Math. **Mathematisch-Physikalische Klasse**, p. 253-297, 1900.

HOLLAND, John Henry. **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence**. [Massachusetts]: MIT Press, 1992.

KAHLMAYER-MERTENS, Roberto Saraiva et al. **Como elaborar projetos de pesquisa: linguagem e método**. Rio de Janeiro: FGV Editora, 2007.

KARAKATIĆ, Sašo; PODGORELEC, Vili. A survey of genetic algorithms for solving multi depot vehicle routing problem. **Applied Soft Computing**, v. 27, p. 519-532, 2015.

KESSON, Malcolm. **Hilbert curve**: concepts & implementation. c2002. Disponível em: <http://www.fundza.com/algorithmic/space_filling/hilbert/basics/>. Acesso em: 20 out. 2017.

KOPFER, Heiko W.; SCHÖNBERGER, Jörn; KOPFER, Herbert. Reducing greenhouse gas emissions of a heterogeneous vehicle fleet. **Flexible Services and Manufacturing Journal**, v. 26, n. 1-2, p. 221-248, 2014.

LAHYANI, Rahma; KHEMAKHEM, Mahdi; SEMET, Frédéric. Taxonomy for rich vehicle routing problems. In: INTERNATIONAL CONFERENCE ON LOGISTICS OPERATIONS MANAGEMENT, IEEE INTERNATIONAL CONFERENCE-GOL'12. 1., 2012.

LAPORTE, Gilbert et al. Classical and modern heuristics for the vehicle routing problem. **International transactions in operational research**, v. 7, n. 4-5, p. 285-300, 2000. Disponível em: <<http://dx.doi.org/10.1111/j.1475-3995.2000.tb00200.x>>.

LAPORTE, Gilbert. Fifty years of vehicle routing. **Transportation Science**, v. 43, n. 4, p. 408-416, 2009. Disponível em: <<https://doi.org/10.1287/trsc.1090.0301>>.

LINDEN, Ricardo. **Algoritmos genéticos**. 2. ed. Rio de Janeiro: Brasport, 2012.

LIU, Ran; JIANG, Zhibin; GENG, Na. A hybrid genetic algorithm for the multi-depot open vehicle routing problem. **OR spectrum**, v. 36, n. 2, p. 401-421, 2014. Disponível em: <<http://dx.doi.org/10.1007/s00291-012-0289-0>>.

MALAQUIAS, Neli Gomes Lisboa et al. **Uso dos algoritmos genéticos para a otimização de rotas de distribuição**. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Uberlândia, Uberlândia, 2006.

MCPHEE, Nicholas Freitag; POLI, Riccardo; LANGDON, William B. **Field guide to genetic programming**. Lulu Enterprises, 2008.

MENDES, Joao Batista et al. A Hybrid Multiobjective Evolutionary Algorithm for Truck Dispatching in Open-Pit-Mining. **IEEE Latin America Transactions**, v. 14, n. 3, p. 1329-1334, 2016.

MITCHELL, Melanie. **An introduction to genetic algorithms**. [Massachusetts]: MIT Press, 1998.

MOLE, R. H.; JAMESON, S. R. A sequential route-building algorithm employing a generalised savings criterion. **Journal of the Operational Research Society**, v. 27, n. 2, p. 503-511, 1976.

MONTOYA-TORRES, Jairo R. et al. A literature review on the vehicle routing problem with multiple depots. **Computers & Industrial Engineering**, v. 79, p. 115-129, 2015.

MUELDER, Chris; MA, Kwan-Liu. Rapid graph layout using space filling curves. **IEEE Transactions on Visualization and Computer Graphics**, v. 14, n. 6, 2008.

NAZIF, Habibeh; LEE, Lai Soon. Optimised crossover genetic algorithm for capacitated vehicle routing problem. **Applied Mathematical Modelling**, v. 36, n. 5, p. 2110-2117, 2012.

NECULA, Raluca; BREABAN, Mihaela; RASCHIP, Madalina. Performance evaluation of ant colony systems for the single-depot multiple traveling salesman problem. In: INTERNATIONAL CONFERENCE ON HYBRID ARTIFICIAL INTELLIGENCE SYSTEMS. Springer, Cham. p. 257-268, 2015a.

NECULA, Raluca; BREABAN, Mihaela; RASCHIP, Madalina. Tackling the Bi-criteria Facet of Multiple Traveling Salesman Problem with Ant Colony Systems. In: IEEE INTERNATIONAL CONFERENCE ON TOOLS WITH ARTIFICIAL INTELLIGENCE (ICTAI). 27., 2015b. p. 873-880, 2015b.

PLATZMAN, Loren K.; BARTHOLDI III, John J. **Spacefilling curves and routing problems in the plane**. Georgia Inst of Tech Atlanta Production and Distribution Research Center, 1983.

PEANO, Giuseppe. Sur une courbe, qui remplit toute une aire plane. **Mathematische Annalen**, v. 36, n. 1, p. 157-160, 1890.

REBELLO, Fabrício Rocha; HAMACHER, Silvio. Uma Proposta de Algoritmo Genético de Duas Fases para Roteamento de Veículos. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL. 32., 2000.

RIBEIRO, Glaydston Mattos; LORENA, Luiz Antonio Nogueira. Roteamento de veículos dinâmico usando algoritmos genéticos. In: ANPET - Congresso de Pesquisa e Ensino em Transportes. 16., 2005. **Anais...**

RAIMANN, Eliane. **Geração de rotas urbanas virtuais usando algoritmos genéticos**. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Uberlândia, Minas Gerais, 2007.

ROSE, Nicholas J. Hilbert-type space-filling curves. **North Carolina State University**. Chicago, 2001.

SAGAN, Hans. **Space-filling curves**. [S. l.]: Springer Science & Business Media, 2012.

SERGEYEV, Yaroslav D.; STRONGIN, Roman G.; LERA, Daniela. **Introduction to global optimization exploiting space-filling curves**. Springer Science & Business Media, 2013.

TALBI, El-Ghazali. **Metaheuristics: from design to implementation**. [S. l.]: John Wiley & Sons, 2009.

TASAN, A. Serdar; GEN, Mitsuo. A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. **Computers & Industrial Engineering**, v. 62, n. 3, p. 755-761, 2012.

TOTH, Paolo; VIGO, Daniele (Ed.). **Vehicle routing: problems, methods, and applications**. Society for Industrial and Applied Mathematics, 2014.

UCHOA, Eduardo et al. New benchmark instances for the capacitated vehicle routing problem. **European Journal of Operational Research**, v. 257, n. 3, p. 845-858, 2017.

VIDAL, Thibaut et al. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. **Computers & operations research**, v. 40, n. 1, p. 475-489, 2013a.

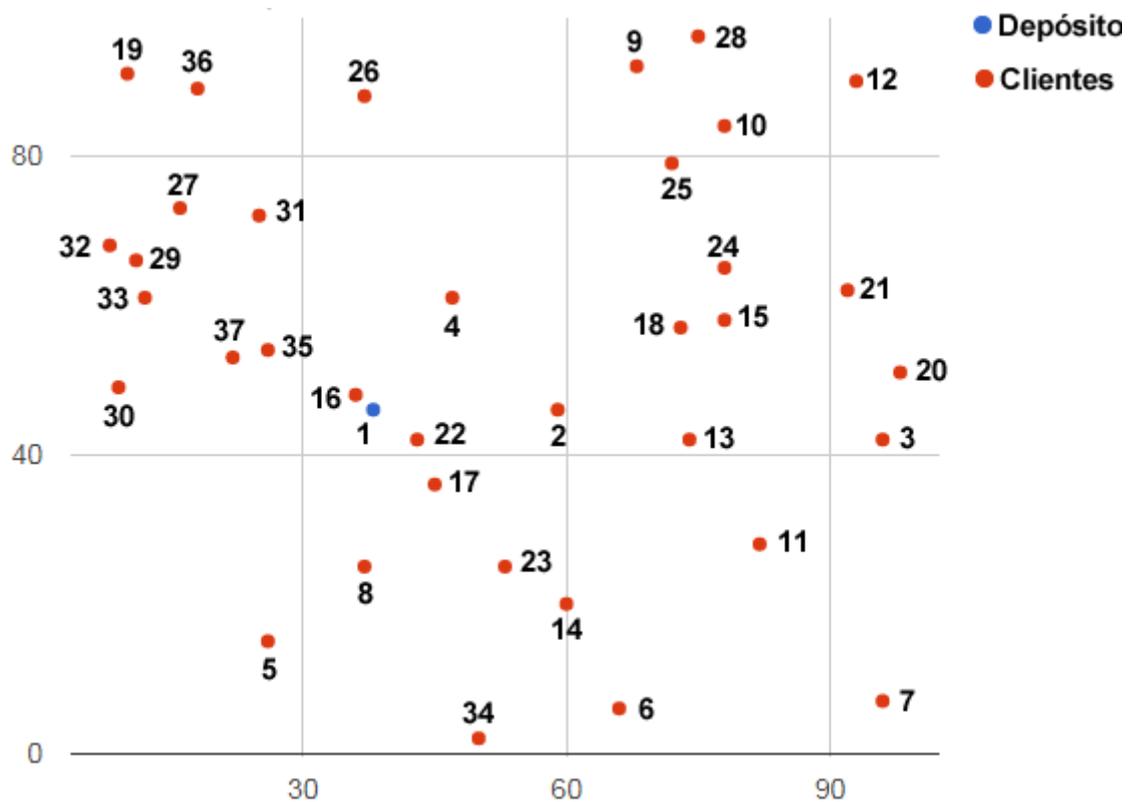
VIDAL, Thibaut et al. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. **European Journal of Operational Research**, v. 231, n. 1, p. 1-21, 2013b.

WU, Luciele; CUNHA, Cláudio Barbieri. O problema da roteirização periódica de veículos. **Transportes**, v. 16, n. 1, 2008.

APÊNDICE A - Comparações dos resultados alcançados

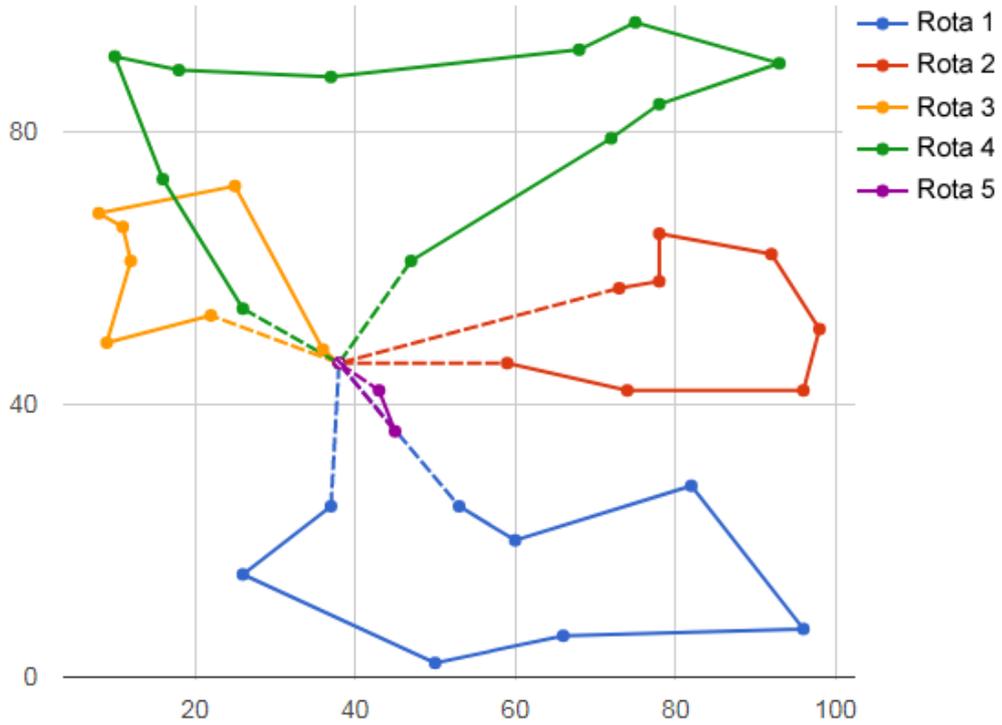
Nesta seção são apresentados alguns resultados de comparações realizadas entre a solução ótima e o algoritmo híbrido proposto neste trabalho para diferentes instâncias de VRPs existentes na literatura.

Figura 1 - Instância A-n37-k5 da base de *benchmark*, espaço de busca contendo 5 veículos e 36 clientes.



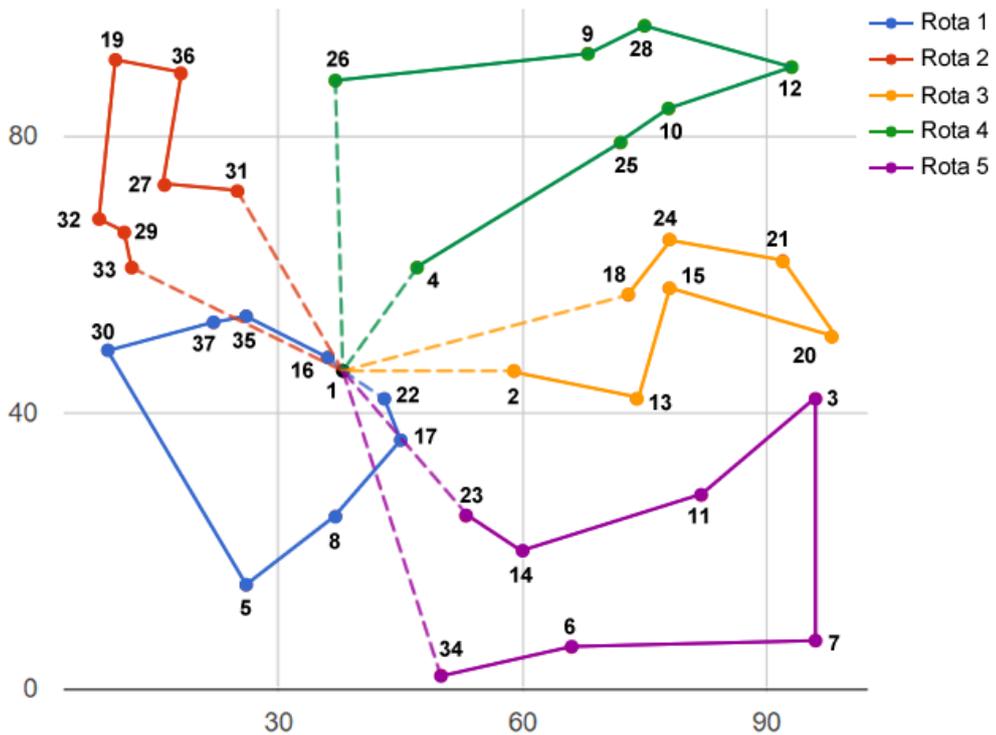
Fonte: Uchoa et al. (2017).

Figura 2 - Solução ótima contendo custo de 669, apresentada por Uchoa (2017).



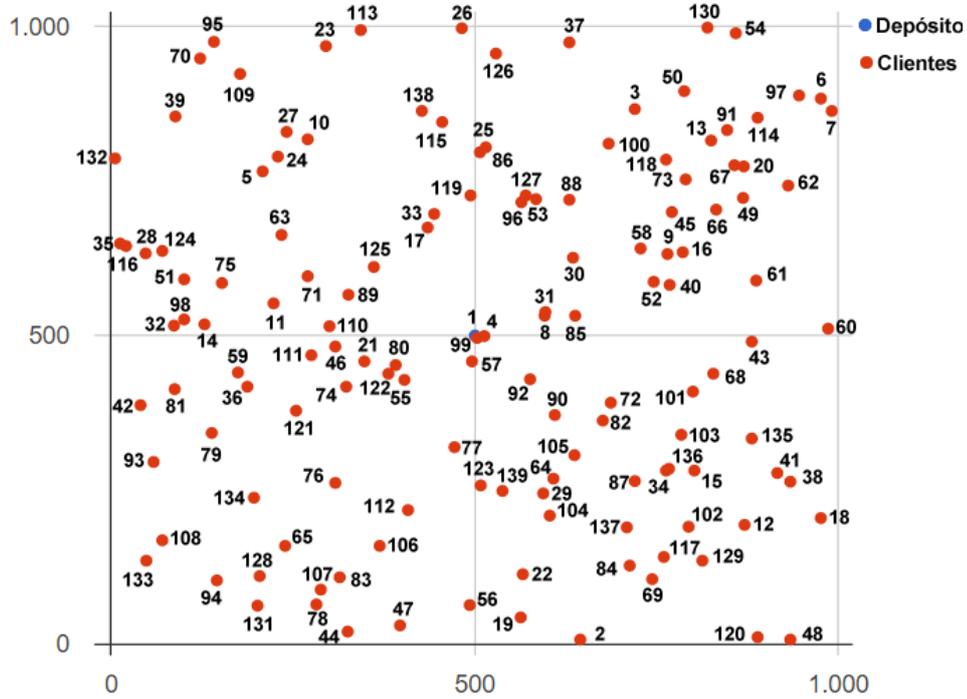
Fonte: Uchoa et al. (2017).

Figura 3 - Resultado do algoritmo híbrido contendo custo de 756, uma diferença de 13% da solução ótima.



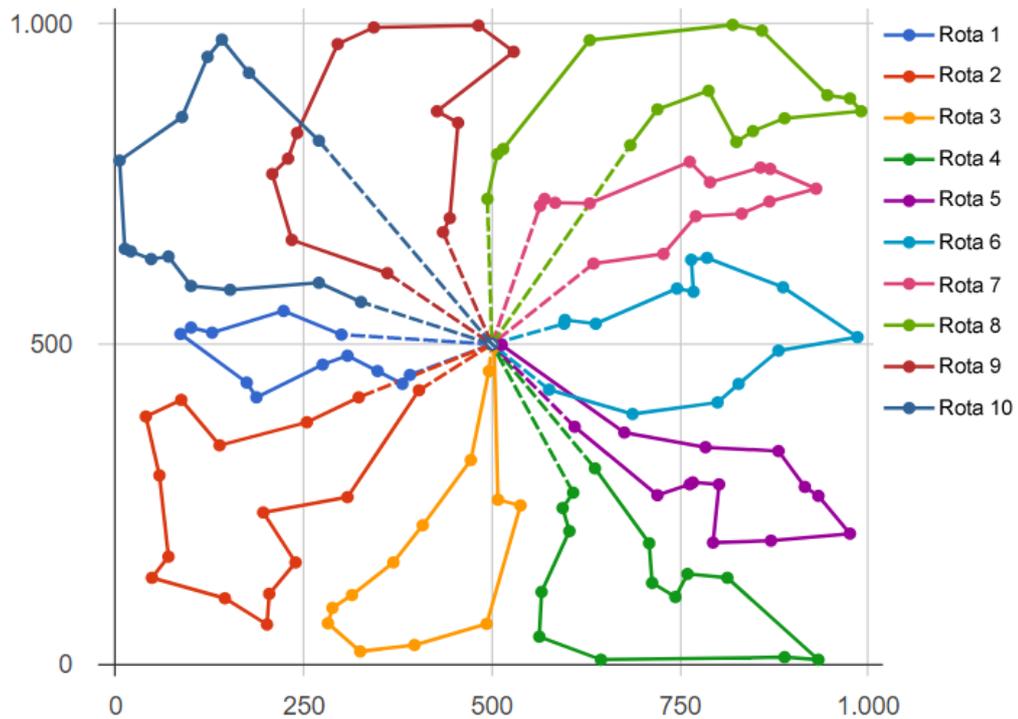
Fonte: Próprio autor.

Figura 4 - Instância X-n139-k10 da base de *benchmark*, espaço de busca contendo 10 veículos e 138 clientes.



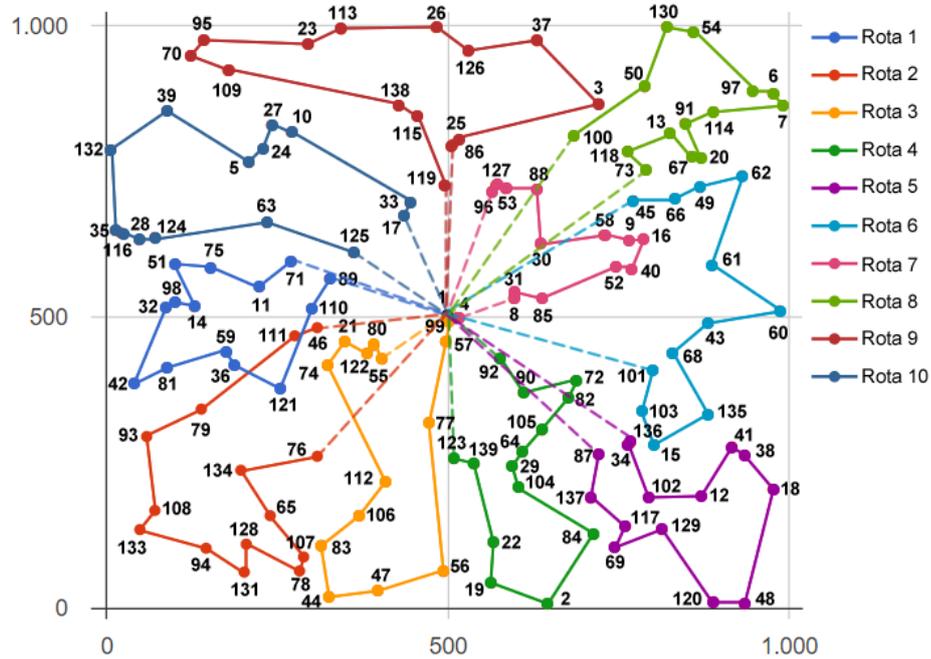
Fonte: Uchoa et al. (2017).

Figura 5 - Solução ótima contendo custo de 13590, apresentada por Uchoa et al. (2017).



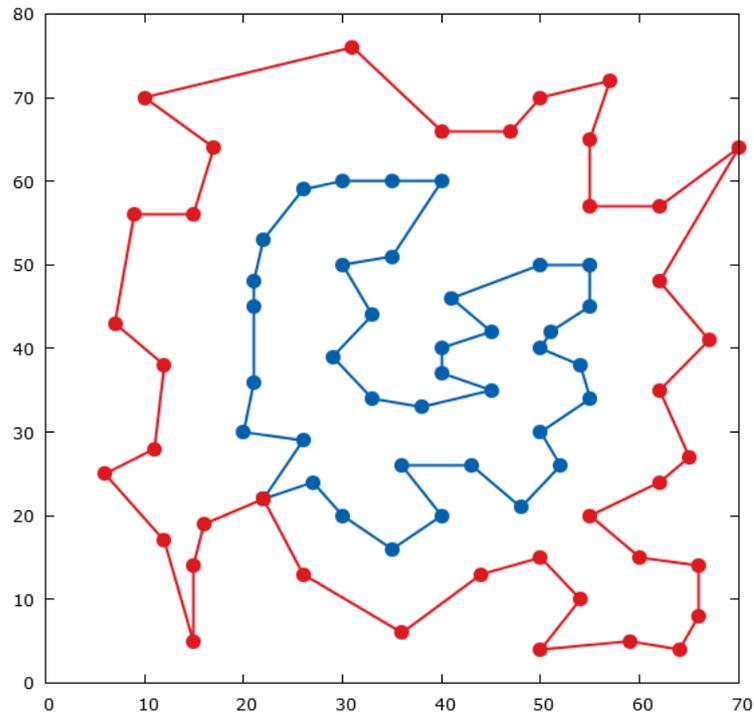
Fonte: Uchoa et al. (2017).

Figura 6 - Resultado do algoritmo híbrido contendo custo de 14840, uma diferença de 9,2% da solução ótima.



Fonte: Próprio autor.

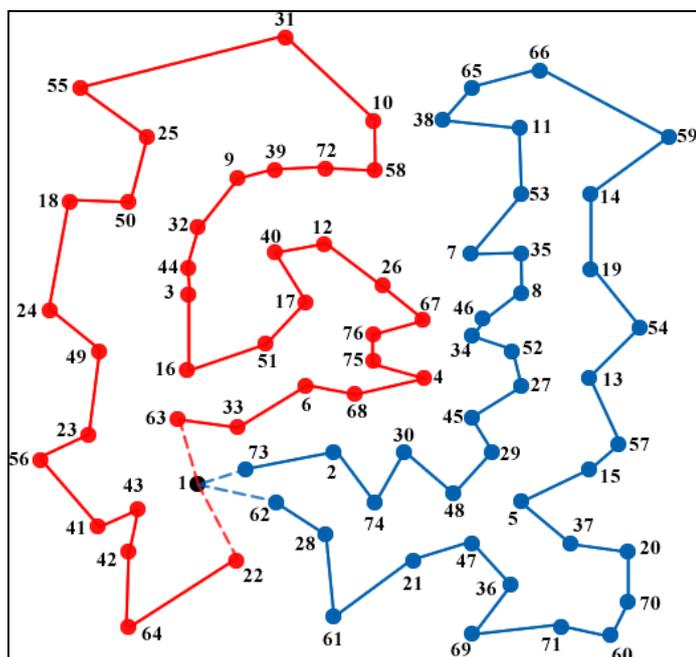
Figura 7 - Instância eil76 da base de *benchmark*, contendo 2 veículos e 75 clientes.



Resolução proposta por Necula, Breaban e Raschip (2015a) com custo final de 558,59

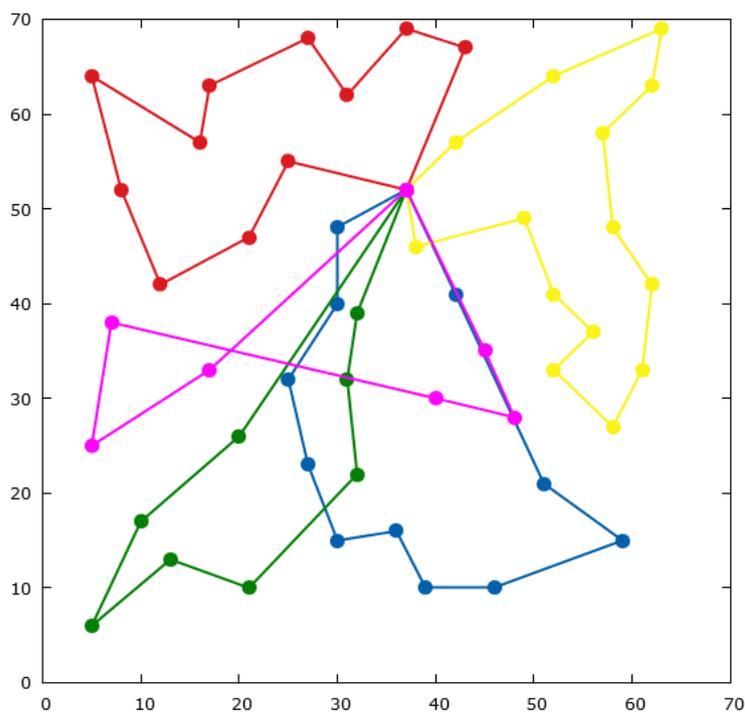
Fonte: Necula; Breaban; Raschip (2015a).

Figura 8 - Resultado do algoritmo híbrido proposto com custo final de 566,88, uma diferença de 1,5% da solução ótima.



Fonte: Próprio autor.

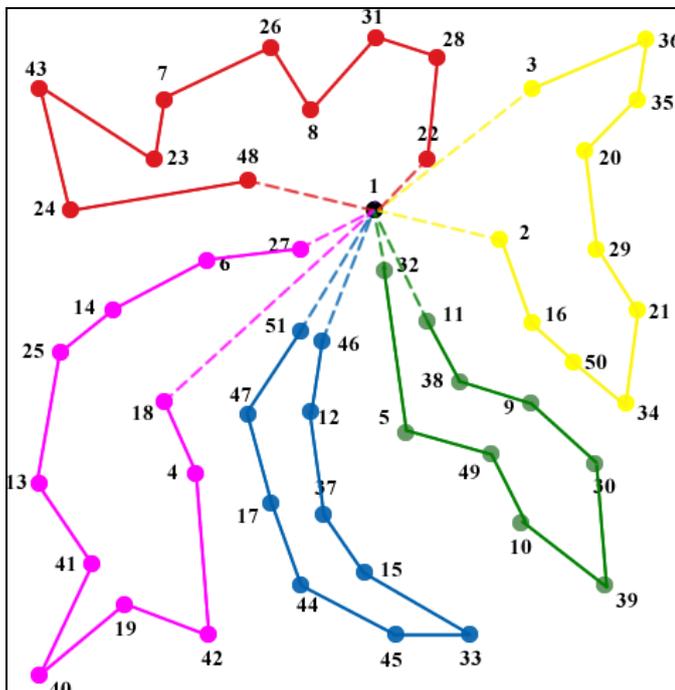
Figura 9 - Instância eil51 da base de *benchmark*, contendo 5 veículos e 50 clientes.



Resolução proposta por Necula, Breaban e Raschip (2015a) com custo final de 615,19.

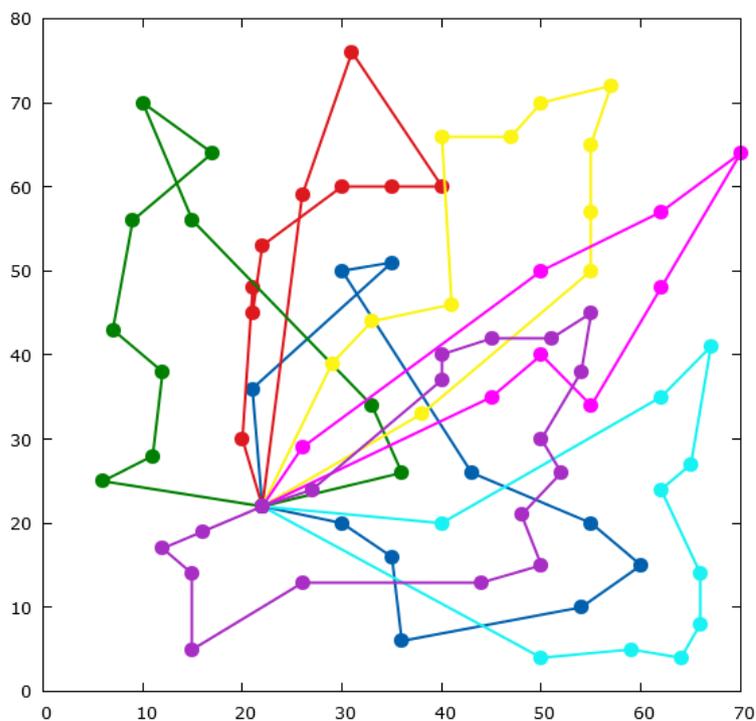
Fonte: Necula; Breaban; e Raschip (2015a).

Figura 10 - Resultado do algoritmo híbrido proposto com custo final de 557,75, uma diferença de 7,4% da solução ótima.



Fonte: Próprio autor.

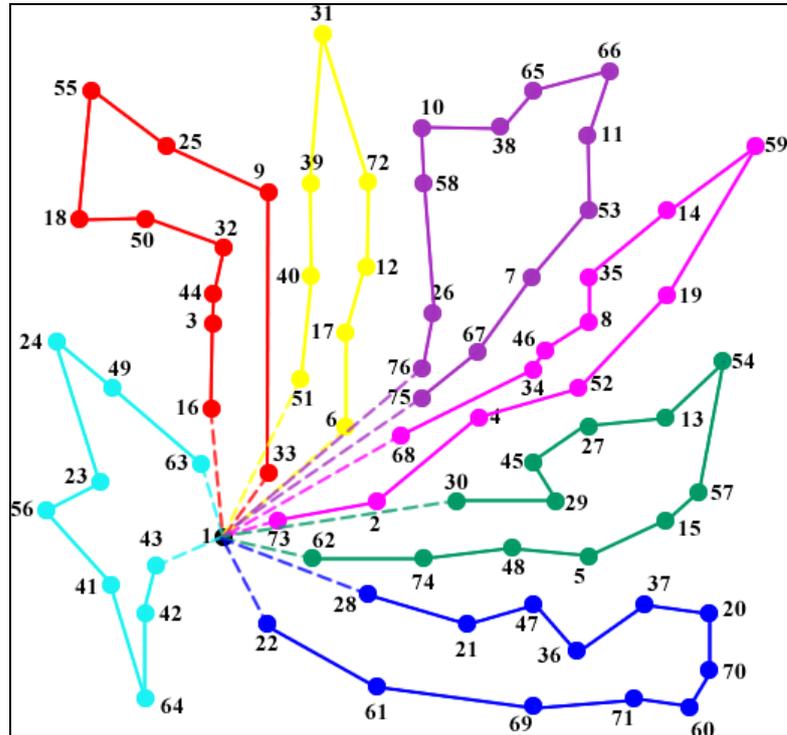
Figura 11 - Instância eil76 da base de *benchmark*, contendo 7 veículos e 75 clientes.



Resolução proposta por Necula, Breaban e Raschip (2015a) com custo final de 964,69.

Fonte: Necula; Breaban; e Raschip (2015a).

Figura 12 - Resultado do algoritmo híbrido proposto com custo final de 817,48, uma diferença de 7,6% da solução ótima.



Fonte: Próprio autor.

APÊNDICE B - Grafos

A teoria dos grafos é um ramo da matemática que estuda as relações entre os objetos de um determinado conjunto. Para tal são empregadas estruturas chamadas de grafos, $G(V, E)$, onde V é um conjunto não vazio de objetos denominados vértices e E um subconjunto de pares não ordenados de V , chamados arestas.

Dependendo da aplicação, arestas podem ou não possuir direção. Em um grafo, pode ser permitido ou não arestas serem ligadas a mais de um vértice, que, por sua vez, ainda pode possuir um peso (numérico) associado. E caso as arestas possuam uma direção associada (indicada por uma seta na representação gráfica), então temos um dígrafo (grafo orientado).

Os grafos são geralmente representados graficamente da seguinte maneira: é desenhado um círculo para cada vértice, e para cada aresta é desenhado um arco conectando suas extremidades. Entretanto, convenções alternativas para a representação de grafos podem ser adotadas, que incluem representações por adjacência como empacotamento de círculos, onde vértices são representados como regiões disjuntas no plano e arestas são representadas por adjacências entre regiões.

Na área da Inteligência Artificial, muitos algoritmos heurísticos representando o espaço de busca do problema em forma de grafos, com o objetivo de representar seus estados de forma organizada.

A seguir são apresentados os principais tipos de grafos existentes na literatura (GOLDBARG; LUNA, 2005):

- Grafo simples: é um grafo não direcionado, sem laços e com, no máximo uma aresta entre quaisquer dois vértices (sem arestas paralelas). Para um grafo simples, o número de vizinhos de um vértice é igual à sua valência.

- Grafo completo: é o grafo simples em que, para cada vértice do grafo, existe uma aresta conectando este vértice a cada um dos demais, ou seja, todos os vértices do grafo possuem mesmo grau. Um grafo completo de n vértices é frequentemente denotado por K_n . Ele tem $n(n-1)/2$ arestas (correspondendo a todas as possíveis escolhas de pares de vértices).
- Grafo nulo: é o grafo cujo conjunto de vértices é vazio.
- Grafo vazio: é o grafo cujo conjunto de arestas é vazio.
- Grafo trivial: é o grafo que possui apenas um vértice e nenhuma aresta.
- Grafo regular é um grafo em que todos os vértices possuem o mesmo grau.
- Multigrafo: é um grafo que permite múltiplas arestas ligando os mesmos vértices (arestas paralelas).
- Pseudografo: é um grafo que contém arestas paralelas e laços.
- Grafo direcionado ou orientado: um grafo é dito direcionado quando o sentido das ligações entre os vértices é importante. Quando a família de arestas é formada por pares ordenados dizemos que o grafo é orientado ou direcionado. Nesse caso, ao desenhar o grafo, devemos usar uma seta partindo do primeiro elemento do par até o segundo.
- Grafo ponderado: Um grafo $G = (V, E)$ é ponderado se existem valores numéricos associados a suas arestas ou nós.
- Grafo rotulados: Um grafo $G = (V, E)$ é rotulado se existem atribuições associados a seus nós (tanto numéricas como alfabéticas).
- Grafo planar: um grafo G é dito planar se puder ser representado graficamente no plano de tal forma que não haja cruzamento de suas arestas.

Caso contrário o grafo é dito não-planar. É um tipo de grafo muito importante por representar os problemas de topologia plana. Redes de estradas, mapas, circuitos eletrônicos, vários tipos de layouts etc, quando representados via grafo, acabam configurando, na maioria das ocasiões, como grafos planares.

- Grafo conexo: um grafo é conexo se for possível estabelecer um caminho de qualquer vértice para qualquer outro vértice de um grafo. Se for sempre possível estabelecer um caminho de qualquer vértice para qualquer outro vértice mesmo depois de remover $k - 1$ vértices, então diz-se que o grafo está k -conexo. Note que um grafo está k -conexo se, e somente se, contém k caminhos independentes entre qualquer par de vértices. Em um grafo genérico G , o corte associado a um conjunto X de vértices é o conjunto de todas as arestas que têm uma ponta em X e outra em $V(G) - X$, onde $V(G)$ é o conjunto de todos os vértices pertencentes ao grafo G .
- Árvore é um grafo simples acíclico e conexo. Às vezes, um vértice da árvore é distinto e chamado de *raiz*. As árvores são muito usadas como estruturas de dados na área da computação.