

UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MANOEL SARMENTO LIMA NETO

PROPOSTA PARA OTIMIZAÇÃO DE ROTAS DE ENTREGA DE
MERENDA ESCOLAR NA REDE PÚBLICA DA CIDADE DE MANAUS

MANAUS
2017

UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MANOEL SARMENTO LIMA NETO

PROPOSTA PARA OTIMIZAÇÃO DE ROTAS DE ENTREGA DE
MERENDA ESCOLAR NA REDE PÚBLICA DA CIDADE DE MANAUS

Dissertação apresentada ao Curso de Mestrado em Engenharia Elétrica, área de concentração Controle e Automação de Sistemas do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas.

Orientador: Prof. Dr. Cícero Ferreira Fernandes Costa Filho
Co-Orientadora: Prof^a. Dr^a. Marly Guimarães Fernandes Costa

MANAUS
2017

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

L732p Lima Neto, Manoel Sarmento
Proposta para otimização de rotas de entrega de merenda escolar na rede pública da cidade de Manaus / Manoel Sarmento Lima Neto. 2017
121 f.: il. color; 31 cm.

Orientador: Cícero Ferreira Fernandes Costa Filho
Coorientador: Marly Guimarães Fernandes Costa
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Amazonas.

1. Problema de Roteamento de Veículos. 2. Otimização de rotas. 3. Recozimento Simulado. 4. Método de geração de vizinhos. 5. Algoritmo de busca A Star. I. Costa Filho, Cícero Ferreira Fernandes II. Universidade Federal do Amazonas III. Título

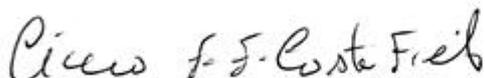
MANOEL SARMENTO LIMA NETO

PROPOSTA PARA OTIMIZAÇÃO DE ROTAS DE ENTREGA DE MERENDA ESCOLAR NA REDE PÚBLICA DA CIDADE DE MANAUS.

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

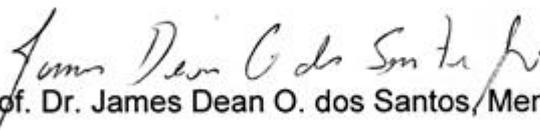
Aprovado em 28 de março de 2017.

BANCA EXAMINADORA



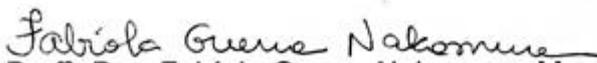
Prof. Dr. Cicero Ferreira Fernandes Costa Filho, Presidente

Universidade Federal do Amazonas- UFAM



Prof. Dr. James Dean O. dos Santos, Membro

Universidade Federal do Amazonas- UFAM



Prof.^a Dra. Fabiola Guerra Nakamura, Membro

Universidade Federal do Amazonas- UFAM

AGRADECIMENTOS

Primeiramente, a Deus pelo dom da vida e por me conceder a oportunidade de estudar.

Aos meus orientadores, Prof. Dr. Cícero Ferreira Fernandes Costa Filho e Profa. Dra. Marly Guimarães Fernandes Costa;

Aos meus pais, Manoel Sarmiento Lima Filho e Maria Gerlane de Aquino Lima, à minha irmã, Carolyn de Aquino Lima, à minha namorada, Ana Maria Bezerra de Souza, e a todos os meus familiares pelo apoio;

Aos meus colegas de turma Arthur, Carmina, Andrews, Francisco, João Paulo, Jonilson, Larissa, Márcio e Robson que sempre estiveram me incentivando e ajudando no desenvolvimento do trabalho;

Aos professores do curso de pós-graduação em engenharia elétrica pelo aprendizado adquirido;

À Universidade Federal do Amazonas e em especial ao Centro de Tecnologia Eletrônica e da Informação – CETELI - pela concessão de toda infra-estrutura para a realização desse trabalho;

Aos irmãos da Igreja Batista Regular Elion por suas orações;

À secretária de educação do município de Manaus, Kátia Helena Serafina Cruz Schweickardt;

Ao diretor do departamento de engenharia da secretaria municipal de educação de Manaus, Hudson Breno;

À diretora do departamento de tecnologia da informação da secretaria municipal de educação, Rosalina Lobo, e ao gerente de sistema Ailton da Cruz Galvão;

Ao diretor do departamento de logística da secretaria municipal de educação, Leís da Silva Batista, por me disponibilizar os dados que foram utilizados para o desenvolvimento deste trabalho.

RESUMO

O problema de roteamento de veículos é importante quando falamos de logística, tanto da logística de uma empresa quanto da administração dos recursos públicos. Esta dissertação propõe uma abordagem híbrida para resolver o Problema de Roteamento de Veículos com Janela de tempo (*VRPTW*). Para avaliar esta nova abordagem, um estudo de caso foi proposto com o objetivo de determinar as rotas de entrega de merenda escolar na rede pública de ensino da cidade de Manaus – Amazonas, Brasil. A solução otimizada deve minimizar a distância percorrida pelo veículo de entrega, atendendo as restrições de capacidade e de janela de tempo. A abordagem apresentada é do tipo global local e foi denominada de método global-local-g. Onde, na busca global utiliza-se a meta-heurística recozimento simulado com uma nova técnica de geração de vizinhos, denominada geração de vizinhos por quadrante. A busca local tem a finalidade de refinar todas as soluções encontradas pela busca global. Para isso utilizou-se o algoritmo de busca A^* em conjunto com uma nova heurística, denominada heurística do passo seguinte. A grande diferença entre a abordagem apresentada e outras encontradas na literatura é que nesta nova abordagem toda a solução encontrada pelo método global é otimizada através de buscas locais. Na literatura, em trabalhos semelhantes, as distâncias entre os nodos de interesse, nesse caso as escolas, já são pré-calculadas ou expressas em termos da distância Euclidiana entre esses nodos. Na dissertação ora apresentada, os cálculos de distância entre os nodos são realizados em tempo de execução e levam em conta a distância de logradouro (distância que considera o percurso das ruas em um mapa). Na busca global, o método recozimento simulado, utilizando o método geração de vizinhos por quadrante, gera soluções formadas por uma sequência contendo os nodos de interesse. Utilizando então, a distância de logradouro e as restrições da janela de tempo e de capacidade dos veículos, são definidos agrupamentos de nodos. Em seguida, dentro de cada um dos agrupamentos, otimiza-se o percurso percorrido através de buscas locais, algoritmo A^* com a heurística do passo seguinte. Os resultados obtidos são comparados com outras abordagens apresentadas no trabalho. Essas comparações levam em consideração o tempo computacional e a distância total percorrida. A abordagem desenvolvida apresentou resultados relevantes em comparação com as outras abordagens, tanto em tempo de execução, quanto em distância percorrida.

Palavras-chave: Problema de roteamento de veículos; Otimização de rotas; Recozimento Simulado; Método de geração de vizinhos; Algoritmo de busca A Star.

ABSTRACT

The vehicle routing problem is critical when it comes to company logistics and the administration of public resources. This work proposes a hybrid approach to solve the Vehicle Routing Problem with Time Window (VRPTW). In order to evaluate this new approach, a case study was proposed with the goal to establish the school meal delivery routes in the public school system of Manaus city, Brazil. Particularly, the optimized solution (route) should minimize the distance traveled by the delivery vehicle, taking into account both capacity and time window constraints. In addition, the proposed approach is the global-local kind and it was named global-local-g method. In the global search, the metaheuristic simulated annealing (SA) is combined with a new technique to generate neighbors, named neighbors' generation per quadrant. In contrast, the main purpose of the local search is the refinement of all solutions found by the global search. For this reason, the A* algorithm is combined with a new heuristic, called as heuristic of the next step. It is worth noticed that the main difference between the approach presented here to others in the literature is the local optimization. In this new approach, all solutions from the global method are optimized through local searches. In similar works, the distances between the nodes of interest (*i.e.*, schools) are already pre-calculated in terms of the Euclidean distance between these nodes. However, in this work, the distance between the nodes is calculated at run time and it takes into account the distance from the actual streets, *i.e.*, it considers the route distance between streets in a map. In the global search, the SA method combined with the neighbors' generation per quadrant produces all solutions, which are formed by a sequence of the nodes of interest. Then, using the street distances, window time restrictions, and the vehicle capacity, all node clusters are defined. Thus, within each cluster, the path traveled through local searches is optimized by the A* algorithm combined with the heuristic of the next step. Experimental results with the proposed approach presented prominent results in comparison to the other existing ones regarding run time and distance traveled.

Keywords: Vehicle Routing Problem; Route Optimization; Simulated Annealing; Neighbor Generation Method; A* Algorithm.

LISTA DE ILUSTRAÇÕES

Figura 1 – Custos referentes ao custo logístico.	11
Figura 2 – Representatividade dos custos logísticos do Brasil em relação ao PIB.	12
Figura 3 – Mapa do município de Manaus com a região de estudo demarcada.	14
Figura 4 – Interessados na logística voltada para cidades (Taniguchi, 2001)	19
Figura 5 – Categorização do <i>VRP</i> cidade na literatura (Kim <i>et al.</i> , 2015)	20
Figura 6 – Variantes da pesquisa sobre <i>VRP</i> ao longo do tempo. (Kim <i>et al.</i> , 2015).....	22
Figura 7 - Representação gráfica de uma solução viável para o CluVRP (Pop e Chira, 2014).	32
Figura 8 – Grafo e dígrafo. (a) Um grafo não direcionado $G = (V, E)$, onde $V = \{1, 2, 3, 4, 5, 6\}$ e $E = \{1, 2, 1, 5, 2, 5, 3, 6\}$. O vértice 4 é isolado. (b) Um dígrafo $G = (V, E)$, onde $V = \{1, 2, 3, 4, 5, 6\}$ e $E = \{1, 2, 2, 2, 2, 4, 2, 5, 4, 1, 4, 5, 5, 4, 6, 3\}$. O arco (2, 2) é um laço ou auto-loop. (c) O subgrafo do grafo da parte (a) inferido pelo conjunto de vértices $\{1, 2, 3, 6\}$	45
Figura 9 – Grafo não direcionado $G = (V, E)$, onde $V = \{1, 2, 3, 4, 5\}$ e $E = \{1, 2, 2, 1, 1, 5, 5, 1, 2, 5, 5, 2, 2, 4, 4, 2, 2, 3, 3, 2, 3, 4, 4, 3, 4, 5, 5, 4\}$. (a) representação gráfica. (b) representação por lista de adjacências. (c) representação por matriz de adjacências.	49
Figura 10 - Grafo direcionado $G = (V, E)$, onde $V = \{1, 2, 3, 4, 5, 6\}$ e $E = \{1, 2, 1, 4, 2, 5, 3, 5, 3, 6, 4, 2, 5, 4, 6, 6\}$. O arco (6, 6) é um laço ou auto-loop.(a) representação gráfica. (b) representação por lista de adjacências. (c) representação por matriz de adjacências.	50
Figura 11 – Grafo dirigido e valorado $G(V, E, P)$. (a) representação gráfica. (b) representação em matriz de adjacências (ou matriz de valores), vértices de origem são ligados aos vértices de destino com os pesos atribuídos na célula que os conecta. (c) representação em lista de adjacências, um exemplo seria o vértice 0 é conectado ao vértice 1 com peso 3 e ao vértice 5 com peso 1.....	50
Figura 12 – Algoritmos de busca e suas subdivisões adaptado de (Souza, 2008).....	52
Figura 13 – Pseudocódigo do algoritmo Dijkstra.....	55
Figura 14 – Exemplo de uma busca gulosa com a função heurística: $f(n)=h(n)$	57
Figura 15 – Fluxograma básico do algoritmo A estrela.	59
Figura 16 – Exemplo de uma busca A estrela com função heurística $f=g+h$	60
Figura 17 – Fluxograma básico do algoritmo de recozimento simulado.	65
Figura 18 – Dados Georreferenciados.	68
Figura 19 – Fluxograma mostrando as principais características do Método Proposto e dos métodos. O termo por quadrante refere-se ao método de geração de vizinhos por quadrante. 71	71
Figura 20 – (a) Representações utilizadas no algoritmo de Busca Global – Recozimento Simulado. (b) Representação utilizada nos algoritmos de Busca Local – Algoritmo de busca A* ou Dijkstra	75
Figura 21 – Coordenadas polares e cartesianas de uma escola	76
Figura 22 – Região IV dividida em quadrantes.	77
Figura 23 – Fluxograma de geração de vizinhos por quadrantes no método SA.....	80
Figura 24 – Rotina para manter os parâmetros (<i>rmédio</i> , <i>θmédio</i>) no interior da área definida.....	81
Figura 25 – Rotina para atribuir as escolas aos quadrantes.....	81
Figura 26 – Rotina para determinar o quadrante de início do <i>tour</i>	81
Figura 27 – Rotina para determinar a orientação de giro de <i>tour</i>	82

Figura 28 – Sequência em que os quadrantes são percorridos em função do giro e do quadrante de início: (a) origem no quadrante 1, (b) origem no quadrante 2, (c) origem no quadrante 3, (d) origem no quadrante 4.....	83
Figura 29 – Solução para T=1, Geração=1, Quadrante de início =1, Orientação de giro = Anti-horário Diagonal Horário, [0 4 5 7 9 8 12 6 13 15 11 10 14 3 1 2]......	85
Figura 30 – Solução para T=1, Geração=2, Quadrante de início =1, Orientação de giro = Anti-horário Diagonal Horário, [0 4 5 8 12 6 9 7 10 14 3 1 2 13 15 11]......	86
Figura 31 – Solução para T=1, Geração=3, Quadrante de início =1, Orientação de giro = Anti-horário Diagonal Horário, [0 4 5 7 9 8 12 6 11 15 13 10 14 3 1 2]......	86
Figura 32 – Solução para T=0,5, Geração=1, Quadrante de início =2, Orientação de giro = Horário Diagonal Anti-horário, [0 6 9 7 12 8 4 5 10 14 3 1 2 13 15 11].	87
Figura 33 – Solução para T=0,5, Geração=2, Quadrante de início =2, Orientação de giro = Horário Diagonal Anti-horário, [0 6 9 7 8 12 4 5 10 14 3 1 2 13 15 11].	88
Figura 34 – Solução para T=0,5, Quadrante de início =4, Orientação de giro = Anti-horário, Geração=3, [0 6 9 7 8 12 4 5 10 14 3 1 2 13 15 11]......	89
Figura 35 – T=1, Geração=1, Rota1 = [0 4 5 7 9 8 12 0]......	93
Figura 36 –T=1, Geração=1, Rota2 = [0 6 0]......	93
Figura 37 – T=1, Geração=1, Rota3 = [0 13 15 11 10 0]......	94
Figura 38 – T=1, Geração=1, Rota4=[0 14 3 1 2 0]......	94
Figura 39 –T=1, Geração=2, Rota1 = [0 4 5 8 12 6 9 0]......	95
Figura 40 –T=1, Geração=2, Rota2 = [0 7 0]......	96
Figura 41 –T=1, Geração=2, Rota3 = [0 10 14 3 1 0]......	96
Figura 42 –T=1, Geração=2, Rota4 = [0 2 13 15 0]......	97
Figura 43 –T=1, Geração=2, Rota5 = [0 11 0]......	97
Figura 44 – Solução para T=1, Geração=3, Rota1=[0 4 5 7 9 8 12 0]......	98
Figura 45 – Solução para T=1, Geração=3, Rota2=[0 6 0].	99
Figura 46 – Solução para T=1, Geração=3, Rota3=[0 11 15 13 10 0]......	99
Figura 47 – Solução para T=1, Geração=3, Rota4=[0 14 3 1 2 0]......	100
Figura 48 – Fluxograma do algoritmo de busca A* com a heurística do passo seguinte.....	101
Figura 49 – Fluxograma da heurística do passo seguinte.....	101
Figura 50 – Exemplo de grafo para a heurística do passo seguinte.....	102
Figura 51 – Exemplo de grafo para a heurística do passo seguinte após a retirada do Nodo B por esse não possuir vizinhos do passo seguinte.	103
Figura 52 – Grafo final após a aplicação da heurística do passo seguinte.	103

LISTA DE TABELAS

Tabela 1 – Funções e objetivos dos diversos atores na teia urbana (Taniguchi, 2001).....	20
Tabela 2 – Objetivos e Métodos de Modelagem (Kim <i>et. al.</i> , 2015).	25
Tabela 3 – Métodos de Modelagem e Métodos de Solução do VRP (Kim <i>et. al.</i> , 2015).	26
Tabela 4 - Principais artigos abordados na revisão bibliográfica.....	42
Tabela 5 – Componentes da definição formal de um problema.....	51
Tabela 6 – Vizinhos e vizinhos do passo seguinte no nodo A	102
Tabela 7 – Resultados obtidos com experimento 1.	106
Tabela 8 – Resultados obtidos com experimento 2.	108
Tabela 9 – Resultados obtidos com experimento 3 - busca local: Algoritmo A*.....	110
Tabela 10 – Resultados obtidos com experimento 3 - busca local: Algoritmo Dijkstra.	111
Tabela 11 – Resultados obtidos com experimento 4.	112

LISTA DE SIGLAS

ACO	Ant Colony Optimization
AFS	Average Feasibles Solutions
AFV	Alternative Fuel Vehicle
BB	Branch and Bound
BFS	Breadth First Search
BRA	Block Recombination Algorithm
BS	Best Solution
CX	Cycle Crossover
CFS	Cost First Search
CluVRP	Clustered Vehicle Routing Problem
CVRP	Capacitated Vehicle Routing Problem
DFS	Depth First Search
ER	Edge Recombination
FIFO	First In First Out
FRX	Fragmente Reordering Crossover
GA	Genetic Algorithm
GBFS	Greedy Best First Search
GIS	Geographic Information System
GLG	Global Local G
GS	Greedy Search
GVRP	Green Vehicle Routing Problem
HVRP	Heterogeneous Vehicle Routing Problem
ILS	Iterated Local Search
IP	Integer Programming
LIFO	Last In First Out
MDVRP	Multiple Depot Vehicle Routing Problem
MDCVRP	Multiple Depot Capacitated Vehicle Routing Problem
MIP	Mixed Integer Programming
NP-Hard	Non Deterministic Polynomial Time Hard
OBM	Order Based Mutation
OX	Order Crossover
OVRPSTW	Open Vehicle Routing Problem with Soft Time Windows
PBM	Position Based Mutation
PBX	Position Based Crossover
PDP	Pickup and Delivery Problem
PMX	Partially Mapped Crossover
QNG	Quadrant Neighbor Generation
SA	Simulated Annealing
SDVRP	Single Depot Vehicle Routing Problem
SEMED	Secretaria Municipal de Educação
SM	Scramble Mutation
SS	Scatter Search
TS	Tabu Search
TSVCG	Tabu Search Algorithm with Variable Cluster Grouping
VNS	Variable Neighborhood Search
VRP	Vehicle Routing Problem
VRPTW	Vehicle Routing Problem with Time Window

SUMÁRIO

INTRODUÇÃO	11
1.1 OBJETIVOS	17
1.1.1 Objetivo Geral	17
1.1.2 Objetivos específicos.....	17
1.2 ORGANIZAÇÃO.....	18
REVISÃO BIBLIOGRÁFICA	19
2.1 PESQUISA BIBLIOGRÁFICA.....	21
2.2 PROBLEMAS DE ROTEAMENTO DE VEÍCULOS (VRP'S)	21
2.3 MÉTODOS PARA RESOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS (VRP'S)	25
2.4 PUBLICAÇÕES DA ÁREA DE ROTEAMENTO DE VEÍCULOS.....	29
2.4.1 Algoritmo Genético (GA)	29
2.4.2 Busca Tabu (TS)	32
2.4.3 Recozimento Simulado (SA)	35
2.4.4 Algoritmo de Varredura com Algoritmo Guloso	37
2.4.5 GENILS (GENIUS, ILS e VND)	38
FUNDAMENTAÇÃO TEÓRICA.....	44
3.1 GRAFOS.....	44
3.1.1 Definição.....	44
3.1.2 Dígrafo ou grafo direcionado	45
3.1.3 Incidência	45
3.1.4 Adjacência.....	46
3.1.5 Grau	46
3.1.6 Caminho e comprimento.....	47
3.1.7 Subcaminho	47
3.1.8 Circuito, ciclo, e, laço.....	47
3.1.9 Grafo conexo, e componentes conexos	48
3.1.10 Grafo fortemente conexo, e componentes fortemente conexos	48
3.1.11 Grafo valorado	48
3.1.12 Representações de grafos	49
3.2 AGENTES DE RESOLUÇÃO DE PROBLEMAS	51
3.3 BUSCA SEM INFORMAÇÃO (BUSCA CEGA).....	52
3.3.1 Busca em largura (<i>Breadth-First-Search – BFS</i>)	53
3.3.2 Busca em profundidade (<i>Depth-First-Search – DFS</i>)	53
3.3.3 Dijkstra (<i>Cost-First-Search – CFS</i>).....	54
3.4 BUSCA COM INFORMAÇÃO (HEURÍSTICA).....	55

3.4.1 Função Heurística	56
3.4.2 Busca gulosa pela melhor escolha (<i>Greedy Best-First-Search – GBFS</i>).....	57
3.4.3 Busca A estrela (<i>A Star Search – A*</i>).....	58
3.5 ALGORITMO HEURÍSTICO GLOBAL	60
3.5.1 Recozimento Simulado (<i>Simulated Annealing - SA</i>).....	60
Função objetivo	61
Critério de Metrópolis	61
Parâmetros	62
Estruturas de vizinhança	63
Fluxograma básico	65
MATERIAIS E MÉTODOS	67
4.1 MATERIAIS	67
4.2 MÉTODOS	70
4.2.1 Problema de roteamento de veículos com janela de tempo.....	72
4.2.2 Codificação da solução	74
4.2.3 Geração de vizinhos por quadrante.....	76
Divisão em quadrantes	76
Atribuição das escolas aos quadrantes	78
Geração de vizinhos	78
4.2.4 Função objetivo	89
4.2.5 Heurística do passo seguinte.....	100
4.2.6 Plataforma de desenvolvimento	104
RESULTADOS E DISCUSSÕES.....	105
5.1 ANÁLISE DE DESEMPENHO DOS MÉTODOS	105
5.2 EXPERIMENTO 1: COMPARAÇÃO ENTRE OS MÉTODOS GL E GLG	105
5.3 EXPERIMENTO 2: COMPARAÇÃO ENTRE OS MÉTODOS GL E QNG	107
5.4 EXPERIMENTO 3: COMPARAÇÃO ENTRE OS MÉTODOS QNG E QNG_GLG	109
5.5 EXPERIMENTO 4: COMPARAÇÃO ENTRE O MÉTODO QNG_GLG E O MÉTODO PROPOSTO	112
CONCLUSÕES	114
REFERÊNCIAS	117

INTRODUÇÃO

A Secretaria Municipal de Educação do Município de Manaus (SEMED) vem tendo um alto custo logístico com a entrega de merenda escolar para escolas da rede pública de ensino. Esse quadro se agravou quando em 4 de março de 2013 deu-se início ao Programa Alimentar, criado através da parceria da prefeitura de Manaus com a SEMED. O mesmo trouxe aos alunos da rede pública de ensino um aumento no número de refeições proporcionadas no decorrer dos dias letivos. Onde antes existia somente uma refeição, agora passaria a ter até sete refeições diárias, dependendo da modalidade de ensino.

O problema descrito no parágrafo anterior pode ser melhor entendido quando se conhecem os custos que compõem o custo logístico total, a Figura 1 mostra como os custos logísticos são divididos.

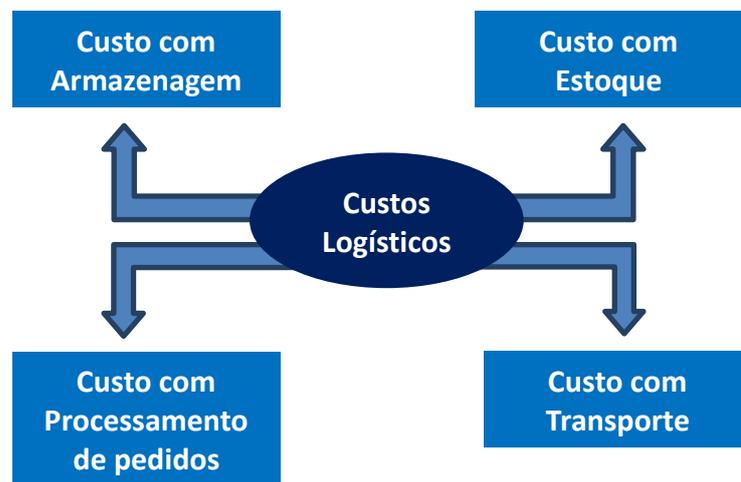


Figura 1 – Custos referentes ao custo logístico.

A soma dos custos apresentados no diagrama da Figura 1 corresponde ao custo logístico. A fim de avaliarmos a parcela com maior influência, é necessário analisarmos o

gráfico mostrado na Figura 2, onde vemos a relação do custo logístico com o PIB do Brasil, no decorrer dos anos de 2004 a 2012.

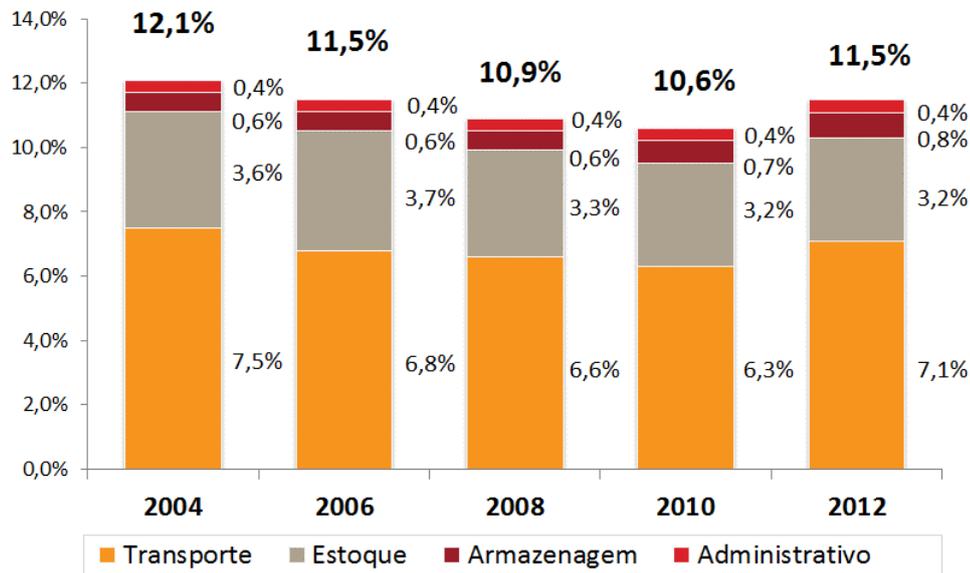


Figura 2 – Representatividade dos custos logísticos do Brasil em relação ao PIB.
Fonte: (Lima e Lobo, 2014).

O gráfico nos mostra que entre os custos que compõem o custo logístico, o custo com o transporte é significativamente maior que os outros. Em (Lima e Lobo, 2014) verificou-se que até 2010, o custo logístico do Brasil em relação ao PIB vinha sofrendo sucessivas quedas, interrompidas em 2012 pela falta de infraestrutura, levando a um retrocesso de seis anos em termos de custos logísticos. O desbalanceamento da matriz de transportes de carga, ao contribuir para o aumento dos gastos das empresas com logística, impactou na economia do Brasil como um todo.

Na China, de acordo com a Federação Chinesa de Logística e Compras Logísticas, na contabilização dos custos envolvidos com a operação básica de logística nacional, a estatística em 2008 mostrou que os custos logísticos foram de 5.500 bilhões de *yuan*, sendo que 2.867 bilhões de *yuan* são referentes aos custos com transporte. Esse valor representa cerca de 52,6% dos custos logísticos (Zhibin *et al.*, 2010).

Assim, podemos afirmar que o transporte desempenha um papel importante das atividades concernentes à logística, pois envolve a aquisição e manutenção de uma frota de veículos e o gasto com combustíveis e pessoal. É essencial que, a partir de um depósito, possa-se suprir a demanda dos clientes com eficiência, a fim de que esse custo diminua.

Reconhecendo o importante papel desenvolvido pelo setor de transporte, em 1959, Dantzig e Ramser introduziram o Problema de Roteamento de Veículos. O *VRP* (do inglês *Vehicle Routing Problem*) veio para contribuir na redução dos custos de logística das empresas e da sociedade.

De acordo com (Dantzig e Ramser, 1959) o problema enquadra-se bem na área de otimização combinatória, e, pode ser posto nos seguintes termos: considerando um conjunto de clientes dispersos, um depósito centralizado e uma frota de veículos que capta bens desse depósito e distribui para os clientes, como determinar um plano de rotas a fim de minimizar o custo de transporte total, ou as distâncias totais. Por causa de suas múltiplas variáveis, com um enorme número de possibilidades de combinação entre as mesmas, é considerado um problema NP-Difícil (*NP-Hard* do inglês *non-deterministic polynomial-time hard*), portanto não há algoritmo em tempo polinomial para encontrar soluções ótimas.

Embora seja um problema que data da década de 50 do século passado, verifica-se, até os dias atuais, um crescimento contínuo de pesquisas nessa área, na busca de métodos mais rápidos e que produzam melhores resultados.

O foco da pesquisa desse trabalho é a roteirização da entrega da merenda escolar numa região específica da área urbana de Manaus. Em 2015, a população do município de Manaus era de 2.057.709 habitantes, o que a colocava na posição de sétima cidade mais populosa brasileira. Sendo que 99,49 % dos habitantes viviam em área urbana e 0,51 % em área rural (IBGE, 2015).

Em 2010, o (IBGE, 2010) constatou-se que, entre as capitais estaduais brasileiras, Manaus foi a que registrou o maior crescimento populacional nas últimas décadas, 22,24%. A taxa de fecundidade era de 2,1 filhos por mulher e mais da metade da população, 67,69% da mesma tinha entre 15 e 64 anos de idade. Cerca de 3,75% da população de Manaus se encontrava em situação de extrema pobreza e sua taxa de analfabetismo na idade de 15 a 24 anos era de 1,2%. O (Ibge, 2015) mostrou que em 2010 a maior parte da população encontrava-se nas zonas norte e leste da cidade, sendo a Cidade Nova o bairro mais populoso, com 121.135 habitantes. Assim, as zonas leste e norte da cidade de Manaus apresentam atualmente a maior concentração de pessoas. Essas zonas formam a região chamada de região IV, que possui a maior concentração de alunos, com 54 escolas divididas em quatro níveis de ensino. Por esse motivo a região IV, situado na confluência dessas duas zonas, mostrada na Figura 3, será a área utilizada para os experimentos sobre roteirização desse trabalho.

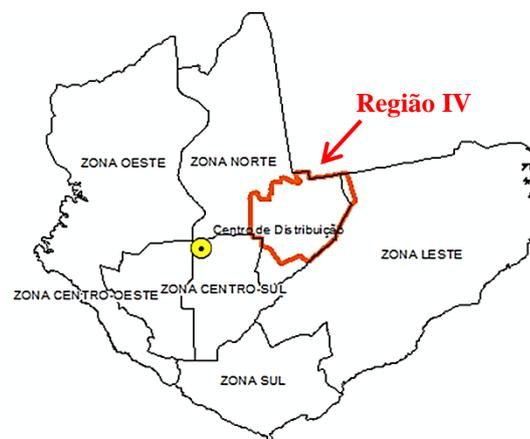


Figura 3 – Mapa do município de Manaus com a região de estudo demarcada.

Em (Marques *et al.*, 2013), foi realizado o primeiro trabalho voltado para a roteirização do transporte da merenda escolar das escolas municipais urbanas de Manaus usando *GIS (Geographic Information Systems)*. Na época em que o trabalho foi realizado, as rotas empregadas para a distribuição da merenda eram determinadas empiricamente, cabendo ao motorista à responsabilidade de definir os melhores percursos. Um problema enfrentado era a frequência de visitas às escolas. A falta de planejamento, além do aumento nos custos de

transporte, ocasionava a falta de suprimentos e, conseqüentemente, falta de alimento para os estudantes. Os autores utilizaram o Software *ArcGIS Desktop 10.1/Arc MAP* para realizar a estruturação da base de dados referente ao cenário da cidade de Manaus. Com a ferramenta *Network Analyst* simulou-se as rotas de entrega da região IV, já apresentada anteriormente na Figura 3. O trabalho foi dividido em três etapas, sendo a primeira a estruturação da base de dados, seguida da análise de grupos, utilizando a técnica de vizinho mais próximo, e a última a roteirização. A iniciativa dos autores contribuiu com o setor de logística, descartando o empirismo e contribuindo para elaboração de um planejamento para a utilização dos veículos existentes para a consecução da tarefa, através de um processo de agrupamento das escolas em rotas de entrega. Entretanto, o trabalho não se preocupou com a capacidade de cada veículo e a demanda das escolas onde seriam realizadas as entregas programadas. Observa-se também que não foram desenvolvidas técnicas de otimização para a criação dos grupos de escolas para a realização das entregas. Os autores informam que a quantidade de grupos foi calculada matematicamente, mas não detalham esse cálculo. Na roteirização deveria ter sido aplicada uma técnica para se encontrar o menor caminho, mas a mesma não é descrita no artigo.

O trabalho ora proposto tem por escopo o desenvolvimento de uma nova metodologia para otimização das rotas utilizadas na entrega da merenda escolar na região IV da cidade de Manaus, anteriormente citada. A principal característica desse trabalho é associar, no processo de otimização, uma busca global e uma busca local. Na busca global é utilizada a técnica de recozimento simulado. Na busca local é utilizado o algoritmo A*. Uma solução encontrada pelo algoritmo de recozimento simulado é constituída por uma lista de escolas. Essa lista é então dividida em grupos (*clusters*) ou rotas de entrega, em função da demanda das mesmas e da capacidade de entrega dos veículos (problema capacitado) ou em função do tempo do percurso (problema de janela de tempo). Dentro de cada um dos grupos de escola aplica-se o

algoritmo A^* para otimização da rota. A **função objetivo** é então calculada somando-se as distâncias das rotas entre os grupos e dentro de cada grupo. É mister frisar que, para cada candidato gerado pela busca global, realizam-se tantas buscas locais quantos forem os grupos de escola formados. Esse método de busca foi denominado nesse trabalho de método global-local-g. A letra g representa o número de buscas locais realizada para cada busca global e corresponde ao número de grupos de escolas formados em função das limitações de capacidade e tempo.

Para o algoritmo de busca local A^* , com o objetivo de acelerar o processo de busca, esse trabalho propõe, uma nova heurística, a **heurística do passo seguinte**. Para determinação da solução inicial e dos vizinhos na técnica de recozimento simulado, foi proposta uma nova técnica de geração de vizinhos, a **técnica de geração de vizinhos por quadrante**.

Como expresso no parágrafo anterior, esse trabalho agrega duas abordagens distintas utilizadas em problemas *VRP*: O problema de roteamento de veículos Capacitados (*CVRP*) e o Problema de Roteamento de Veículos com Janelas de Tempo (*VRPTW*). Na solução proposta, identificamos três novas contribuições à área: 1. A proposta de um novo método de otimização, o método global-local-g, sendo g otimizações locais acionadas sempre que um candidato é gerado pela técnica de otimização global; 2. A proposta de uma nova técnica para geração de vizinhos para o método de recozimento simulado, a técnica de geração de vizinhos por quadrante; 3. A proposta de uma nova heurística para agilizar a busca A^* , a heurística do passo seguinte. Vale salientar que a associação da busca global a uma busca local já vem sendo utilizado na literatura (Lin, Ying, Lee e Chen, 2006), (Fanggeng *et al.*, 2009), (Hai *et al.*, 2010), (Pop e Chira, 2014). No entanto, nos trabalhos anteriores, após a geração de um candidato pela busca global, não são efetuadas buscas locais nos *clusters*. A busca local é efetuada apenas após o término da busca global. No trabalho presente, buscas pelos menores

caminhos são acionadas repetidas vezes, a cada solução apresentada pela busca global. Todas essas contribuições ficarão mais claras no capítulo de metodologia.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Propor um novo método que usa conceito de busca global e busca local para o Problema de Roteamento de Veículos, o método global-local-g, visando melhorar a geração de rotas de entrega de merenda escolar na rede municipal de ensino.

1.1.2 Objetivos específicos

- No problema de otimização VRP, associar cada candidato da busca local a g buscas locais, em que g corresponde ao número de grupos formados devido as limitações de capacidade ou tempo.
- Na busca global, avaliar o desempenho do algoritmo de recozimento simulado e propor para o mesmo uma nova técnica de geração de vizinhos por quadrante.
- Na busca local avaliar o desempenho do algoritmo A^* e propor para a mesma uma nova heurística para aceleração da busca, a heurística do passo seguinte.
- Na busca local, comparar o desempenho do algoritmo A^* com a heurística do passo seguinte ao desempenho do algoritmo Dijkstra.
- Gerar uma solução para o problema de roteamento de veículos para entrega de merenda escolar na região IV da cidade de Manaus.

1.2 ORGANIZAÇÃO

Essa dissertação de mestrado está dividida nos seguintes capítulos:

- Introdução (Capítulo 1);
- Revisão Bibliográfica (Capítulo 2);
- Fundamentação Teórica (Capítulo 3);
- Materiais e Métodos (Capítulo 4);
- Resultados (Capítulo 5);
- Discussões e Conclusões (Capítulo 6);
- Referências Bibliográficas;

O Capítulo 2 tem como foco a identificação de alguns trabalhos relacionados às varias derivações do problema de roteamento de veículos e técnicas de otimização que contribuíram para a riqueza do nosso trabalho, através de uma pesquisa bibliográfica nas principais bases de dados científicas. No Capítulo 3, apresentamos a fundamentação teórica necessária ao desenvolvimento do presente trabalho através da descrição das ferramentas utilizadas.

No Capítulo 4, apresenta-se o conjunto de dados e os passos metodológicos utilizados no trabalho. No Capítulo 5 são apresentados os resultados obtidos com o método proposto, tais como: desempenho, tempo de execução e distância das rotas. Apresentam-se, também, tabelas comparativas com os demais métodos elencados na dissertação. Por fim, as conclusões do trabalho são apresentadas.

REVISÃO BIBLIOGRÁFICA

O capítulo anterior caracteriza o *VRP* através de conceitos vistos em artigos desenvolvidos na área de otimização e apresenta uma introdução sobre o assunto, mostrando seu contexto histórico. Apresenta também a motivação do trabalho atual e delimita seus objetivos. Neste capítulo, analisamos artigos desenvolvidos na área de *VRP* de uma maneira aprofundada, caracterizando os problemas que foram resolvidos e as soluções aplicadas. Sempre que possível, realizamos comparações com o trabalho ora apresentado.

De acordo com (Taniguchi, 2001) o objetivo da logística na cidade é sistematizar e otimizar a logística dentro de uma área urbana, considerando os custos e benefícios das soluções para os setores público e privado. Conforme mostrado na Figura 4 existem quatro partes interessadas na logística voltada para a área urbana: carregadores, moradores, transportadores e administradores.

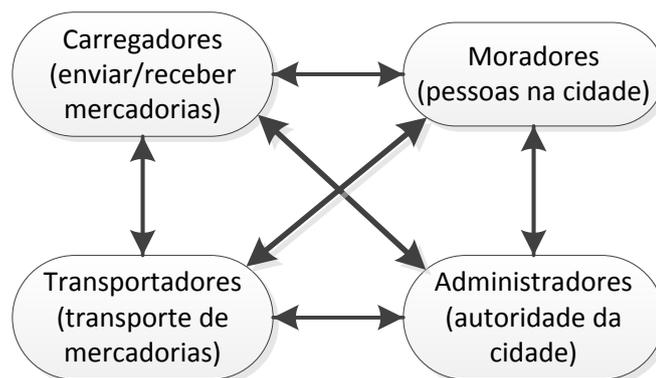


Figura 4 – Interessados na logística voltada para cidades (Taniguchi, 2001)

As setas bidirecionais retratam uma relação de dependência entre todos os atores interessados. Qualquer política implementada por um dos atores interessados deve afetar os

outros, quer através de influências diretas ou impactos indiretos. Cada ator possui um papel distinto. As funções e objetivos de cada ator na teia urbana são mostrados na Tabela 1.

Tabela 1 – Funções e objetivos dos diversos atores na teia urbana (Taniguchi, 2001)

Ator	Funções	Objetivos
Carregadores	Carregar as mercadorias para transportadoras ou outros clientes (Clientes de transportadores)	Maximizar o nível de serviço
Transportadoras	Transportar mercadorias	Minimizar custo de transporte
Moradores	Pessoas que vivem, trabalham e fazem compras na cidade	Minimizar a poluição do ar, poluição sonora, o congestionamento do tráfego e os acidentes perto da área residencial
Administradores	Gerir a Cidade	Melhorar o desenvolvimento econômico da cidade, aliviar o congestionamento do tráfego, melhorar o ambiente dentro da cidade

O trabalho proposto é voltado para minimizar o custo de transporte dos transportadores, propondo uma solução para um *VRP*. A categorização de um *VRP* voltado para área urbana foi apresentada por (Kim *et al.*, 2015), conforme mostrado no diagrama da Figura 5.

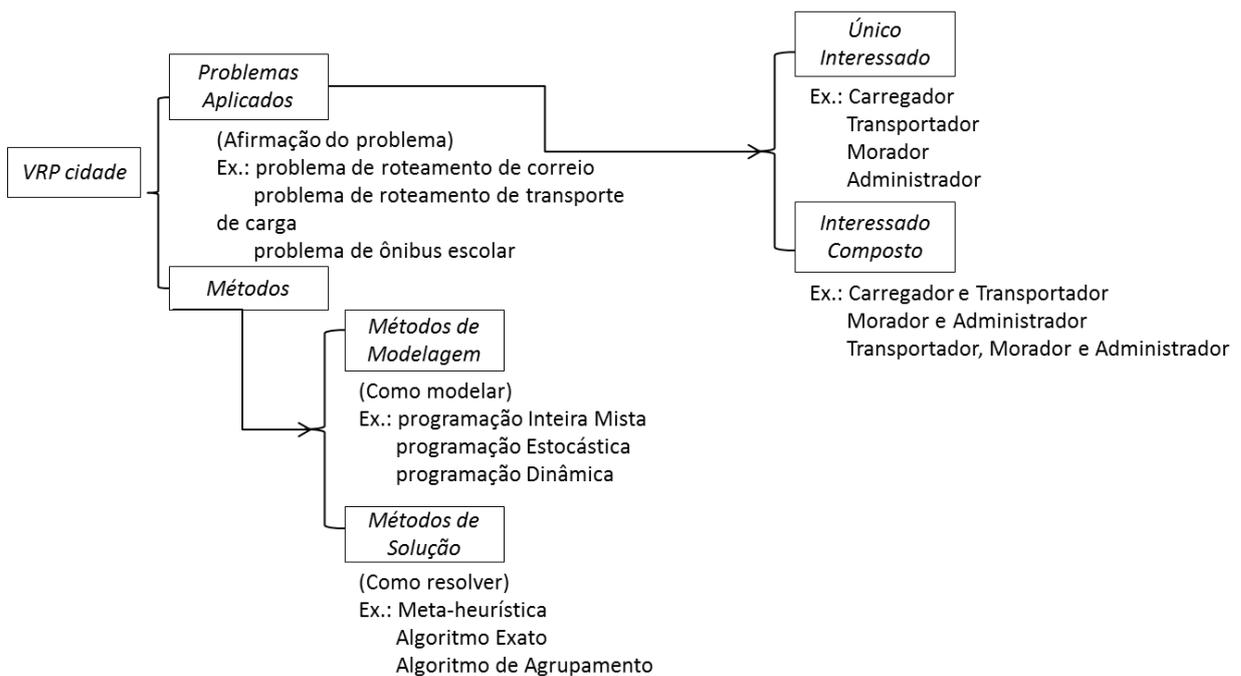


Figura 5 – Categorização do *VRP* cidade na literatura (Kim *et al.*, 2015)

Os problemas de roteamento (*VRP*) mostrados na Figura 5 dizem respeito a um dos atores mostrados na Tabela 1, os transportadores. A solução desse problema impacta diretamente na minimização do custo de transporte. Observando as categorias do diagrama da Figura 5, podemos afirmar que esse trabalho procura resolver um problema aplicado de entrega de merenda escolar, onde o maior interessado é o administrador e o maior beneficiado é a população. A modelagem é realizada através de programação inteira e a solução do *VRP* é obtida através de uma meta-heurística. Neste capítulo abordamos alguns trabalhos que tratam dos problemas de roteamento de veículos já publicados na literatura, mostrando as soluções desenvolvidas tanto para a modelagem quanto para a solução dos mesmos.

2.1 PESQUISA BIBLIOGRÁFICA

Foi realizada uma pesquisa bibliográfica nas seguintes bases de dados científicas: *IEEEXplorer*, *Engineering Village*, *ScienceDirect*, *Management Science* e *InformaPubsOnLine*. Após a leitura inicial, foram selecionados os artigos mais relevantes. Em seguida, fez-se outra busca através dos vocabulários indexados aos mesmos, juntamente com a data de publicação. A partir da análise desses artigos, foram identificados diversos Problemas de Roteamento de Veículos (*VRP's*), com as mais variadas técnicas de modelagem e solução, os quais serão apresentados a seguir.

2.2 PROBLEMAS DE ROTEAMENTO DE VEÍCULOS (*VRP'S*)

Para minimizar o custo com transporte pela primeira vez na década de 60, Dantzig e Ramser (1959) introduziram o Problema de Roteamento de Veículos. Recentemente, em (Kim *et al.*, 2015) o Problema de Roteamento de Veículos (*VRP*) foi caracterizado como uma extensão do problema do Caixeiro Viajante (*TSP – Traveling Salesman Problem*). Dado um

depósito, vários veículos e clientes, o problema é formulado como uma tentativa de responder a seguinte pergunta: como encaminhar os veículos para atender a todos os clientes, minimizando a distância total percorrida?

Ao longo dos anos, conforme mostrado na Figura 6, os pesquisadores, buscando suprir necessidades do cotidiano das cidades, vêm se dedicando a analisar variantes mais específicas desse problema.

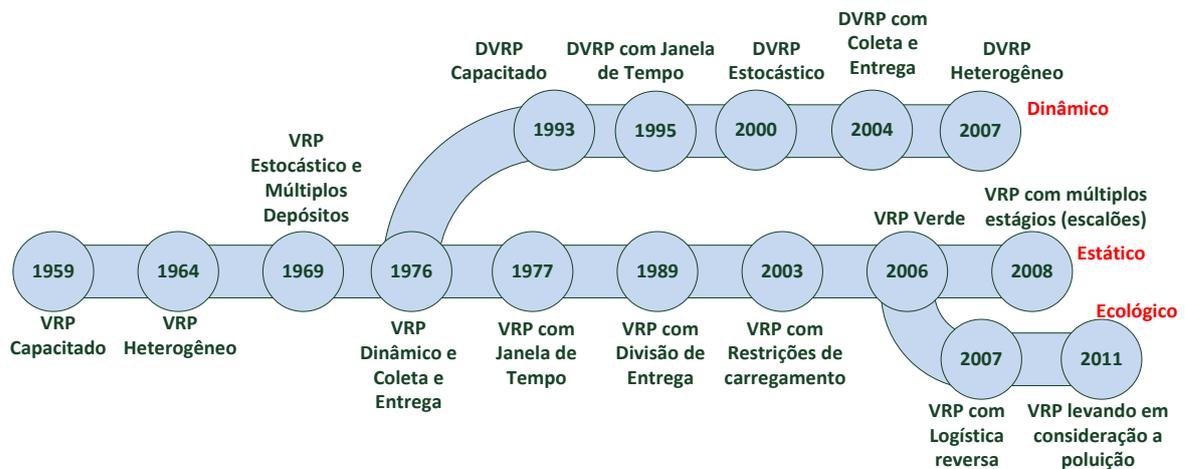


Figura 6 – Variantes da pesquisa sobre VRP ao longo do tempo. (Kim *et al.*, 2015)

Conforme pode ser visto na Figura 6, o ponto de partida dos estudos realizados a partir de 1959 foi o problema *VRP* Capacitado. A partir de 1976 e, ao longo de cinco décadas, foram criados derivados do *VRP*, resultando, atualmente, em três categorias de problemas: Dinâmico, Estático e Ecológico.

O *VRP* Estático é o mais simples e pode ser definido por um grafo $G = (V, \varepsilon, C)$, em que: $V = \{v_0, \dots, v_n\}$ corresponde a um conjunto de vértices; $\varepsilon = \{(v_i, v_j) | (v_i, v_j) \in V^2, i \neq j\}$ corresponde a um conjunto de arcos; e $C = (c_{ij})_{(v_i, v_j) \in \varepsilon}$ corresponde a uma matriz de custo definida por ε , representando distâncias, tempos de viagem, ou custos de viagem. Tradicionalmente, o vértice v_0 é chamado de depósito, enquanto os vértices restantes em V representam os clientes (ou pedidos) que precisam ser atendidos. O *VRP* consiste em

encontrar o conjunto de rotas para K veículos idênticos, baseados no depósito, com cada um dos vértices sendo visitado exatamente uma vez, que minimiza o custo geral da rota (Pillac *et al.*, 2013).

No *VRP* Dinâmico algumas requisições surgem durante a operação dos veículos, e com isso, deve-se realizar uma reprogramação dos veículos em tempo de operação. Como mencionado, no momento da reprogramação das rotas, esse problema pode ser visto como um problema estático a ser resolvido.

Para que as requisições não programadas possam ser atendidas no mesmo dia da solicitação, faz-se necessário impor um tempo limite de solicitação, de tal modo que as demais requisições recebidas após esse tempo limite sejam colocadas para o próximo dia de operação. Pode não existir conhecimento prévio sobre a localização das novas requisições realizadas quando os veículos já estão em operação, e assumi-se existir uma central de comunicação que informa ao motorista mudança na sua programação (Montemanni *et al.*, 2003).

Uma característica que distingue o roteamento dinâmico do estático é encontrada na função objetivo. Enquanto a função objetivo, no contexto estático, minimiza apenas o custo de roteamento, no contexto dinâmico, a mesma inclui outras variáveis, tais como: nível de serviço e rendimento (número de solicitações atendidas por tempo).

O *VRP* Ecológico (*GVRP - Green Vehicle Routing Problem*) introduzido por (Erdoğan e Miller-Hooks, 2012), além de incorporar os objetivos definidos para um *VRP* estático, incorpora também uma preocupação com o meio ambiente e a sociedade. Municípios, agências governamentais, organizações sem fins lucrativos e empresas privadas estão convertendo suas frotas de caminhões para incluir veículos com combustível alternativo (*Alternative Fuel Vehicles – AFVs*), como biodiesel, eletricidade, etanol e hidrogênio, com o objetivo de reduzir voluntariamente o impacto ambiental.

(Uchoa *et al.*, 2006) apresentaram algumas variantes do *VRP* como sendo as mais estudadas pelos pesquisadores:

- Capacitado (*CVRP - Capacitated*) – existe uma frota de veículos com capacidades idênticas, e um único depósito. As demandas são enviadas do depósito para os clientes, sempre respeitando a restrição de capacidade dos veículos. Cada cliente só é visitado por um único veículo. O objetivo do problema é encontrar um conjunto de rotas que apresente um menor custo;
- Janela de Tempo (*VRPTW- Time Window*) – o mesmo problema anterior, com a restrição adicional de que cada cliente só pode ser atendido durante certo horário, chamado de janela de tempo. Essa variante exige que se conheçam os tempos de deslocamento dos veículos, bem como os tempos de atendimento em cada cliente;
- Frota Heterogênea (*HVRP - Heterogeneous*) – podem existir vários tipos de veículos na frota, com diferentes capacidades, tempos de deslocamento e custos operacionais. Normalmente existem as chamadas restrições de docagem, que significam que determinados clientes não podem ser atendidos por certos tipos de veículos;
- Entrega Dividida (*SDVRP – Split Delivery*) – extensão do capacitado em que é permitido que um cliente possa ser atendido por diferentes veículos, se isto possibilitar uma redução do custo da rota;
- Entrega e Coleta (*PDP - Pickup and Delivery Problem*) - pode haver demandas que são coletadas em um determinado cliente para serem entregues em outro cliente. As rotas, porém, ainda devem se iniciar e terminar no depósito;

- Multi-depósito (*MDVRP – Multiple Depot*) - os veículos estão distribuídos por vários depósitos. Eventualmente, uma rota pode começar em um depósito e terminar em outro.

O trabalho ora apresentado tem como base duas dessas variantes, *CVRP* e *VRPTW*, que serão resolvidas através dos métodos descritos no Capítulo 4.

2.3 MÉTODOS PARA RESOLUÇÃO DO PROBLEMA DE ROTEAMENTO DE VEÍCULOS (*VRP'S*)

Na solução de um problema *VRP* dois passos importantes precisam ser dados: a modelagem do problema e a escolha do método para solução do mesmo. Em (Kim *et. al.*, 2015) foi realizada uma revisão tanto das modelagens quanto dos métodos de solução do *VRP* existentes na literatura. A Tabela 2 relaciona o objetivo do problema *VRP* com a modelagem utilizada em diversos trabalhos. A Tabela 3 relaciona o método de modelagem e o método utilizado para solução do *VRP*.

Tabela 2 – Objetivos e Métodos de Modelagem (Kim *et. al.*, 2015).

Objetivos	Métodos de Modelagem
Custo ambiental	Programação Inteira, Programação Inteira Mista, Modelo estocástico e outros.
Custo de viagem	Modelo baseado no conhecimento, Modelo de rede, Programação inteira, Programação inteira mista, Modelo estocástico, Programação dinâmica, outros.
Distância	Programação inteira mista, Programação dinâmica e outros.
Tempo de viagem	Programação inteira, Programação inteira mista, outros.
Custo de unidade	Programação inteira, Programação inteira mista, outros.
Custo fixo	Programação inteira mista.
Tamanho da frota	Programação inteira, Programação inteira mista, outros.
Tempo de serviço	Programação inteira mista.
Janelas de tempo	Modelo espaço tempo
Outras	Modelo de rede, Programação inteira, Programação inteira mista, Modelo espaço tempo, outros.

A modelagem de um *VRP* é feita, na maioria dos casos, através de Programação Inteira (*IP*) ou Programação Inteira Mista (*MIP*). Essa última modelagem caracteriza-se pelo fato de que algumas das variáveis podem não pertencer ao conjunto dos números inteiros.

Tabela 3 – Métodos de Modelagem e Métodos de Solução do VRP (Kim *et. al.*, 2015).

Métodos de Modelagem	Métodos de Solução
Programação Inteira Mista	Algoritmos de Agrupamento, Decomposição, Algoritmos Exatos, Heurística Gulosa, Meta-heurísticas, Novas Heurísticas, Heurística de Remover-Inserir, Algoritmos de Poupança, Heurística de Dois Estágios, Outras.
Programação Inteira	Algoritmos de Agrupamento, Algoritmos Exatos, Heurística Gulosa, Meta-heurísticas, Novas Heurísticas, Algoritmos de Busca, Heurística de Dois Estágios, Outras.
Programação Dinâmica	Algoritmos Exatos.
Modelo Espaço Tempo	Outros.

As principais técnicas de resolução de um *VRP* são: Algoritmos Exatos, Heurísticas Clássicas e Meta-heurísticas, como descrito no trabalho de (Hai *et al.*, 2010). Nos últimos anos, observa-se uma tendência de agrupamento de algumas dessas técnicas, resultando no que podemos denominar de abordagens evolutivas híbridas.

Os métodos exatos são algoritmos em que sempre se consegue uma solução ótima para o problema a ser resolvido. Baseiam-se em técnicas de programação matemática, e raramente resolvem instâncias com mais de 150 clientes (Subramanian *et al.*, 2013).

Alguns Exemplos desses algoritmos são: *Branch and Bound*, proposto por (Land e Doig, 1960), *Cutting Plans* introduzido por (Gomory, 1960), e uma combinação entre esses métodos anteriores, proposta por (Padberg e Rinaldi, 1991), conhecida como *Branch and Cut*.

As heurísticas clássicas desenvolvidas entre 1960 e 1990 fazem parte de uma das classes heurísticas relatadas por (Laporte *et al.*, 2000) em seu trabalho. Estes métodos executam uma exploração relativamente limitada do espaço de busca e, geralmente, produzem soluções (aproximadas) de boa qualidade dentro de tempos de computação modestos. Além

disso, a maior parte das heurísticas clássicas pode ser facilmente adaptada para levar em conta a diversidade das restrições encontradas em contextos práticos.

Exemplos de heurísticas clássicas podem ser: *Savings algorithms* desenvolvidos por (Clarke e Wright, 1964), *Sequential improvement methods* proposto primeiramente por (Mole e Jameson, 1976), *The sweep algorithm* mencionado primeiro por (Wren e Carr, 1971), *Petal algorithms* formulado por (Balinski e Quandt, 1964), *Cluster-first, route-second algorithms* baseado na abordagem de (Miliotis, 1976), e, *improvement heuristics* descrito por (Lin, 1965).

As meta-heurísticas vêm se desenvolvendo nos últimos anos com base em algoritmos inspirados na biologia, física e inteligência artificial. Em (Hai *et al.*, 2010) afirmam que a qualidade das soluções produzidas por esses métodos é geralmente melhor do que a obtida pelas heurísticas clássicas, mas o preço a pagar é o aumento do tempo de processamento. Afinal, a ênfase da meta-heurística é a realização de uma exploração profunda das regiões mais promissoras do espaço de solução. Estes métodos normalmente combinam regras sofisticadas de busca na vizinhança, estruturas de memória e recombinações de soluções.

(Kim *et al.*, 2015) cita alguns dos métodos meta-heurísticos encontrados nas literaturas atualmente: colônia de formigas (*Ant Colony Optimization - ACO*), técnica apresentada por (Dorigo e Gambardella, 1997); algoritmo genético (*Genetic Algorithm - GA*), técnica de otimização e busca fundamentada principalmente por (Holland, 1975); busca em vizinhança variável (*Variable Neighborhood Search - VNS*), introduzida por (Mladenovic, 1995); busca dispersa (*Scatter Search - SS*), introduzida por (Glover, 1977); recozimento simulado (*Simulated Annealing - SA*), introduzido por (Kirkpatrick *et al.*, 1983); busca tabu (*Tabu Search - TS*), introduzida em sua forma atual por (Glover, 1986).

Em (Bai *et al.*, 2015) afirmam que, atualmente, para a solução dos diversos *VRP's*, a maioria dos pesquisadores tende a usar um algoritmo meta-heurístico por atacarem as regiões mais promissoras do espaço de solução. Verifica-se também outra tendência, que é um

interesse crescente no uso de sistema de informação geográfica (*GIS – Geographic Information System*) para analisar problemas dessa natureza.

A fim de melhorar a eficiência das meta-heurísticas, diversas abordagens começaram a adicionar buscas locais, ou fazer combinações com outras técnicas de resolução. Essas são conhecidas como algoritmos evolutivos híbridos, e vêm ganhando grande espaço na atualidade.

(Yen *et al.*, 1998) propuseram uma classificação dos algoritmos evolutivos híbridos em quatro categorias:

- 1) Híbridos *pipelining*: as meta-heurísticas e/ou técnicas de otimização são aplicadas sequencialmente;
- 2) Híbridos assíncronos: utilizam uma população compartilhada com várias técnicas de otimização para prosseguir e cooperar de forma assíncrona.
- 3) Híbridos hierárquicos: usam técnicas de otimização em diversos níveis diferentes de um problema de otimização.
- 4) Algoritmo com operadores adicionais: incluem-se operadores adicionais, como os operadores com movimentos de busca.

Alguns autores na atualidade já utilizaram esses algoritmos aplicados na resolução de diversos tipos de *VRP* (Fanggeng *et al.*, 2009), (Pop e Chira, 2014), e (Santos *et al.*, 2016).

No trabalho ora apresentado, o objetivo principal é encontrar a solução de um *VRP* que apresente a menor distância percorrida. Em prol disso, utiliza-se uma modelagem de programação inteira e um método de solução híbrido que tem como meta-heurística global o recozimento simulado, associado a uma busca local, o algoritmo A*, O recozimento simulado foi a meta-heurística escolhida por ser utilizada em um número considerável de trabalhos apresentados em nossa revisão, e, ainda ser suscetível a integração com um novo método de

geração de vizinhos que será apresentado, Enquanto, o algoritmo de busca A* foi utilizado por apresentar melhores resultados quando aplicado ao mapa utilizado neste trabalho.

2.4 PUBLICAÇÕES DA ÁREA DE ROTEAMENTO DE VEÍCULOS

A área de roteamento de veículos vem se desenvolvendo cada dia mais. Novos problemas estão surgindo à medida que se começa a analisar o dia a dia da logística das empresas e órgãos, sendo necessária a criação de novas soluções, mesmo que essas sejam muitas vezes aproximadas, como aquelas obtidas por heurísticas e meta-heurísticas, conseguem uma redução dos custo de transporte, que é diretamente ligado a logística. Objetivando o entendimento de alguns desses problemas e soluções de maneira mais aprofundada, apresentamos algumas publicações envolvendo abordagens que utilizaram meta-heurísticas como: algoritmos genéticos, busca tabu e recozimento simulado, dentre outras. Algumas das abordagens apresentadas utilizaram outros métodos em conjunto com a meta-heurística, ou seja, são abordagens evolutivas híbridas.

2.4.1 Algoritmo Genético (GA)

No trabalho desenvolvido por (Skok et al., 2000), os autores utilizaram GA em um VRP capacitado com múltiplos depósitos (*MDCVRP – Multiple Depot Capacitated Vehicle Routing Problem*). O problema é decidir que veículos devem entregar para quais clientes, de modo que se obtenha o custo mínimo. A **população inicial** foi gerada de forma randômica. Foram utilizados 4 operadores na fase de **cruzamento**: *PMX (partially mapped crossover)*, *OX (order crossover)*, *CX (cycle crossover)*, *PBX (position based crossover)*, *FRX (fragmente reordering crossover)*, e *ER (edge recombination operator)*. Na fase de mutação foram utilizados 3 operadores: *PBM (position based mutation)*, *OBM (order based mutation)*, e, *SM (scramble mutation)*. O método de **seleção** utilizado, o mesmo que em (Whitley, 1989), possui uma função que depende de um valor denominado tendência, do tamanho da

população, e de um valor randômico entre [0,1]. A **função de aptidão** procura minimizar o custo que envolve a soma do tamanho da rota de cada veículo multiplicado pelo custo unitário de viagem, somados ao número total de veículos utilizados multiplicados pelo custo de aquisição de cada veículo, quando existir a necessidade de se adquirir o veículo novo. A solução do problema apresentou um excelente desempenho em relação ao tempo de processamento, quando o problema envolveu cinco depósitos e 424 clientes, na solução foi apresentado as rotas a serem percorridas pelos veículos e o par de depósitos que cada veículo deveria percorrer.

No trabalho de (Tunjongsirigul e Pongchairerks, 2010), os autores utilizaram *GA* para encontrar a rota de entrega de uma padaria, minimizando a distância total e o número de veículos utilizados. O autor considerou um *VRP* com um único depósito (*SDVRP – Single Depot Vehicle Routing Problem*) e um total de 32 clientes. A **população inicial** foi gerada randomicamente usando o método do vizinho mais próximo. A técnica utilizada para **cruzamento** foi a *PMX*, e a de **mutação** foi a troca recíproca. A técnica de **seleção** utilizada foi a da roleta. As distâncias entre o depósito e os clientes, e de cliente para cliente, foram calculadas pela ferramenta Google Earth. A **função de aptidão** é calculada somando todas as distâncias entre os cromossomos da solução. A técnica desenvolvida mostrou uma melhor solução em número de veículos e distância percorrida do que a apresentada pelo método do vizinho mais próximo. Os resultados apresentados mostraram uma redução de até 50% do número de veículos, e de até 69 km da distância percorrida, em casos com demandas maiores.

No trabalho de (Pop e Chira, 2014) foi desenvolvida uma abordagem híbrida, baseada em algoritmos genéticos (*GA*), para resolução do problema de agrupamento de rotas (*Clustered Vehicle Routing Problem - CluVRP*). O objetivo do trabalho foi o de otimizar o número e a composição dos agrupamentos, de coleta ou entrega. A **população inicial** foi gerada de maneira randômica de modo a respeitar a restrição de capacidade de cada veículo.

Na fase de **cruzamento** utilizou-se o operador *PMX*, e na de **mutação** o operador de troca entre rotas, que realiza a troca randômica entre dois agrupamentos. Na **seleção** de candidatos para reprodução utilizou-se o método da roleta e seleção elitista. O cálculo da **função de aptidão** leva em consideração não só os arcos intra-agrupamentos, como também os arcos inter-agrupamentos. Foi construída uma rede com camadas, de forma que cada camada representa um agrupamento com seus respectivos clientes. A função de adaptabilidade da solução é dada pela soma das distâncias entre todos os agrupamentos que compõem a rede e o valor da soma de todas as rotas locais de todos os agrupamentos. Nesse problema, o depósito é considerado um agrupamento sem cliente. Em cada agrupamento realiza-se, então uma busca local, para determinação do menor caminho. Em cada agrupamento, a distância entre quaisquer dois clientes é predeterminado, ou seja, utiliza-se valores pré-calculados, disponíveis em tabelas de consultas. Na Figura 7 pode-se ver um exemplo de solução utilizando 2 veículos com capacidades idênticas de 25 unidades. As rotas de agrupamentos, também denominadas de rotas globais, estabelecidas pelo algoritmo foram: $(V_0 - V_1 - V_2 - V_0)$ e $(V_0 - V_5 - V_4 - V_3 - V_0)$. Os agrupamentos foram percorridos nessa sequência. As rotas de vértices, também denominadas de rotas generalizadas, foram as seguintes: $(0 - 3 - 2 - 1 - 4 - 5 - 0)$ e $(0 - 11 - 10 - 9 - 8 - 7 - 6 - 0)$.

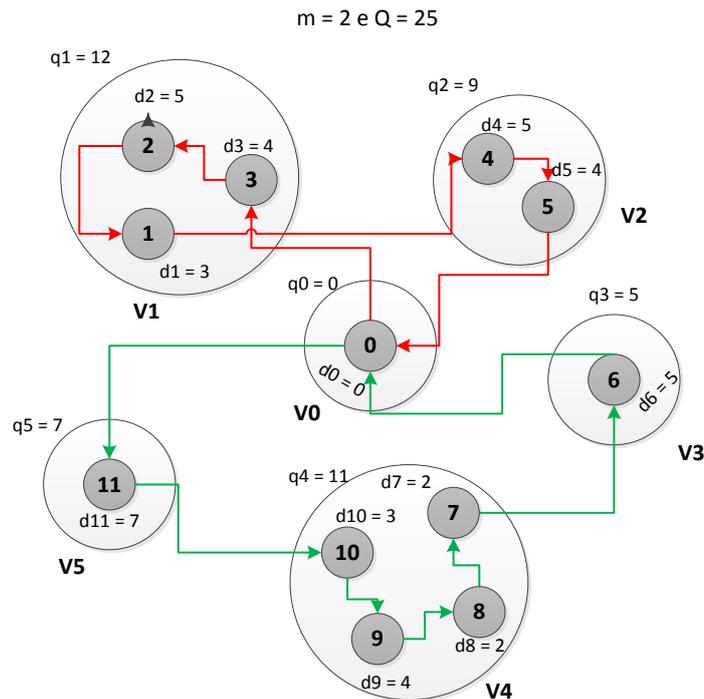


Figura 7 - Representação gráfica de uma solução viável para o CluVRP (Pop e Chira, 2014).

Os experimentos foram realizados considerando várias instâncias do problema *CluVRP* encontradas em (Golden *et al.*, 1998). Nessas instâncias, os agrupamentos foram criados utilizando a abordagem desenvolvida por (Fischetti *et al.*, 1995), onde varia-se o número de vértices, o número de *clusters* e o número de veículos.

Nos resultados computacionais, as melhores soluções foram alcançadas nas primeiras fases das gerações. Por conta disso, o autor afirma que pretende manter a diversidade, variando o tamanho da população e alterando a forma de seleção, com a finalidade de eliminar convergências prematuras. Além disso, pretende otimizar a busca local, a fim de aperfeiçoar as soluções exploradas pelo algoritmo genético híbrido proposto.

2.4.2 Busca Tabu (TS)

Em (Guan *et al.*, 2010) foi utilizada Busca Tabu (*Tabu Search* – TS) para resolver um *VRP*. Primeiramente, os autores obtiveram uma **solução inicial** utilizando um método randômico que não repete clientes, chamado arranjo direto de cliente. O método para geração de **soluções vizinhas** utilizado foi a operação de troca 2-opt. Através do **critério de**

aspiração, menor distância percorrida, é realizada a avaliação da solução gerada. Com a finalidade de fugir de mínimos locais utiliza-se uma **lista tabu** (*taboo list*), que guarda as soluções já encontradas, para que as mesmas não se repitam. Quando o **critério de parada**, um dado número de ciclos, é atingido, espera-se que a solução seja a ótima global. Os testes foram realizados com 10 clientes gerados com coordenadas entre $[0,100]$ e demandas entre $[0,10]$. A comparação realizada pelo autor foi entre a solução inicial gerada e a solução ótima encontrada após o uso da *TS*. Em testes realizados com instâncias geradas pelo próprio autor, foram obtidos os seguintes resultados: uma redução de até 25% no número de veículos e uma redução de aproximadamente 40% na distância percorrida.

O trabalho de (Jinhui *et al.*, 2012) apresentou uma melhoria na *TS*, com a finalidade de resolver o problema de roteamento de veículos aberto (*OVRP*), com janelas de tempo suaves (*Open Vehicle Routing Problem with Soft Time Windows - OVRPSTW*). A principal diferença entre o *OVRP* e o *VRP* é que cada percurso é tratado como um caminho hamiltoniano. A **solução inicial** é criada sequencialmente de forma aleatória, e é considerada uma tabu admissível se não quebrar nenhuma restrição. As **soluções vizinhas** são geradas utilizando o operador de vizinhança 2-opt. Com o intuito de evitar a formação de ciclos (retornar a soluções previamente encontradas) é criada uma **lista tabu dinâmica**. Essa inovação melhora a busca pelo mínimo global. O tamanho da lista é dependente dos valores mínimo (L_{min}) e máximo (L_{max}) encontrados na lista e das quantidades de iterações corrente (n) e máxima (n_{max}).

Os testes foram realizados usando o benchmark do conjunto de dados dos problemas de (Solomon, 1987). Os conjuntos de dados independentes utilizados foram R1, R2, RC1, RC2. Os conjuntos de problemas R1 e R2 são randomicamente distribuídos. Os conjuntos de problemas RC1 e RC2 são randomicamente distribuídos e com uma distribuição de agrupamentos misturados. Os parâmetros utilizados no experimento foram: $L_{min} = 5$,

$L_{max} = 10$, e $n_{max} = 2000$. A vizinhança é de 200. Os resultados foram comparados com os resultados obtidos nos trabalhos de (Repoussis *et al.*, 2007) e o de (Duan, 2009).

A análise de resultados mostra que, quanto ao número de veículos, a *TS* resolveu o problema de maneira ideal e encontrou o resultado de 8 veículos para a classe R1, 2 para a classe R2. Para as classe RC1 e RC2, as quantidades foram 9 e 2, respectivamente. Mais resultados obtidos pelos autores, comparando o custo em termos da distância percorrida e do tempo de execução, podem ser vistos no trabalho em referência.

Em (He et al., 2014), os autores apresentaram uma *TS* com agrupamento de agrupamentos variáveis (*TSVCG – Tabu Search Algorithm with Variable Cluster Grouping*) para lidar com um *VRP* com múltiplos depósitos (*MDVRP - Multiple Depot Vehicle Routing Problem*). Inicialmente, os autores aplicaram uma técnica de juntada de agrupamentos, visando converter o *MDVRP* em um *SDVRP*. A **solução inicial** foi uma distribuição randômica de um veículo para um grupo de clientes. A estrutura de vizinhança construída para gerar as **soluções vizinhas** é composta de 3 operadores, sendo 2 de troca, *1-swap* e *2-opt*, e 1 operador de inserção, *1-insert*. O artigo introduz uma melhoria na técnica de agrupamento dos clientes, separando-os em dois tipos: o primeiro tipo de clientes é constituído por clientes que estão a certa distância de um depósito. O segundo tipo de clientes é constituído por clientes mais distantes do depósito. Um coeficiente de proporção define o limite entre próximo e longe. Através da melhoria nos agrupamentos geram-se melhores soluções. A **função objetivo** a ser minimizada é a distância total percorrida. O *TSVCG (Tabu Search algorithm with Variable Cluster Grouping)* desenvolvido foi aplicado em 23 casos de teste e comparado a trabalhos anteriores. Os autores consideram que o método proposto teve resultados expressivos em termos de redução de tempo de execução e uma melhoria na eficiência ao encontrar o caminho ótimo, especialmente quando o número de clientes é muito grande..

2.4.3 Recozimento Simulado (SA)

O método de Recozimento Simulado (SA – *Simulated Annealing*) também é muito utilizado por pesquisadores da área. No trabalho de (Lin, Ying, Lee e Hsi, 2006) os autores fizeram uso do recozimento simulado para encontrar a solução de um *CVRP*, com o objetivo de minimizar o custo total, que nesse caso é uma função ponderada que engloba o número de veículos e a distância percorrida por cada veículo. O percurso deve ser concebido de tal maneira que cada cliente seja visitado uma vez e por um único veículo. A **solução inicial** foi obtida através do uso de algoritmos de poupança (*savings algorithms*). A **busca global** utilizada para melhoria da solução foi feita com a aplicação de dois operadores de vizinhança são utilizados: o *2-opt exchange* e o *insertion*. Essa abordagem teve como resultado 6 soluções equiparáveis as melhores soluções encontradas na literatura citada no artigo de um total de 14.

No trabalho de (Lin, Ying, Lee e Chen, 2006) foi utilizada a técnica SA para a solução de um *VRPTW*. A **solução inicial** foi gerada através da heurística de inserção sequencial de (Solomon, 1987). Na **busca global** são utilizados dois operadores de vizinhança são utilizados: *2-opt exchange* e *insertion*. Os testes, realizados utilizando como *benchmark* as instâncias de (Solomon, 1987), encontraram todas as melhores soluções nas instâncias C1 e C2. As soluções obtidas nas instâncias R1, R2, RC1 e RC2 foram iguais ou próximas, em torno de 92%, das soluções apresentadas na literatura da época.

Em (Sakakibara *et al.*, 2010) foi considerada a classe de Problema de Roteamento de Veículos com Coleta e Entrega (*Pickup and Delivery Problem - PDP*), com as seguintes restrições: qualquer local que necessite de coleta e entrega é atendido por um veículo em uma rota; o local de coleta é agendado antes do local correspondente de entrega; a capacidade de cada veículo é um dado levado em conta na otimização. A proposta para a resolução desse problema foi uma abordagem híbrida que utiliza técnicas de *branch-and-bound* e a meta-

heurística de recozimento simulado. A chave para a resolução do problema foi decompor o mesmo em subproblemas, pela técnica de divisão (*branch*), com o número de veículos e a quantidade de clientes fixados em cada veículo. Cada subproblema é inserido em um recozimento simulado, onde a **solução inicial** para o subproblema é gerada de forma randômica. Nesse caso, o algoritmo SA é utilizado para realização de uma **busca local** em cada subproblema. Os seguintes operadores de vizinhança são utilizados: *swap* e *re-insertion*. A **função objetivo** utilizada tem por finalidade minimizar uma soma ponderada do tempo de conclusão das coletas e das entregas, com pesos w_1 e w_2 , respectivamente. Uma etapa de limitação fixa os limites inferior e superior para a **função objetivo**, de tal forma a não ser necessário explorar todos os subproblemas. O limite inferior é calculado através de dois métodos, denominadas LB1 e LB2. No método LB1 um modelo relaxado é calculado, retirando todas as restrições binárias e adicionando somente a restrição que determina os clientes pertencentes a cada veículo. No método LB2 usa-se o próprio recozimento simulado para definir o valor aproximado desse limite. O limite superior é calculado somente pelo recozimento simulado.

Os testes foram realizados com 3 instâncias: 5, 10 e 14 clientes, com pesos $w_1 = w_2 = 1$. Foram realizados 10 ensaios, alterando-se as condições iniciais. O melhor plano de transporte foi registrado para cada ensaio.

Os parâmetros do algoritmo SA foram definidos da seguinte maneira: temperatura inicial $T^S = 1,0 \times 10^4$, coeficiente de resfriamento $\delta = 1,0 \times 10^{-6}$ e número de gerações $n = 1,0 \times 10^5$.

Os autores afirmam que as rotas de transporte ótimas (ou próximas das ótimas, com erro de até 1,62%) foram obtidas tanto pelo método híbrido como pelo método SA. No entanto, os resultados obtidos pelo método híbrido foram mais estáveis, encontrou soluções

com o valor das funções objetivos mais próximas durante as iterações, do que pelo método SA.

2.4.4 Algoritmo de Varredura com Algoritmo Guloso

No trabalho de (Chen *et al.*, 2015) foi abordado o Problema de Roteamento de Veículos Capacitados (*CVRP*). Para resolvê-lo foi proposto um algoritmo híbrido que utiliza o algoritmo de varredura (*Sweep Algorithm*) e, posteriormente, uma busca gulosa (*Greedy Search - GS*). A **solução inicial** é obtida ordenando-se os ângulos, coordenadas polares, de cada cliente em relação ao depósito de forma ascendente. Na primeira fase do algoritmo formam-se agrupamentos de clientes. Para isso, inicialmente escolhe-se de forma randômica um cliente. Na sequência, usa-se o método do vizinho mais próximo associado à restrição de capacidade para montar os agrupamentos. Na segunda fase, para melhorar a solução inicial, analisa-se a possibilidade de combinar agrupamentos adjacentes, formando novos agrupamentos, respeitando sempre a restrição de capacidade. A **busca local** foi feita através do algoritmo de busca gulosa. Para a **seleção** de uma nova solução utiliza-se a estratégia de seleção elitista. A **função objetivo** é a soma das distâncias percorridas. Para testar o algoritmo proposto foram utilizados *benchmarks* obtidos a partir de sites internacionais de *VRP*, como (<http://neo.lcc.uma.es/radi-aeb/WebVRP/>). Os resultados foram comparados com (Tiwari e Chang, 2015), que desenvolveram o algoritmo de recombinação de blocos (*Block Recombination Algorithm - BRA*). De acordo com os autores, o método proposto é melhor do que o *BRA*, especialmente em uma instância que utilizou o maior número de veículos, 19 e com 484 nodos, a distância percorrida foi 1137 metros. A taxa de erro referente à solução ótima descrita no *benchmark*, obtido de websites internacionais de problema de roteamento de veículos, foi de 19,86%, enquanto que com o método *BRA* foi de 54,36%.

2.4.5 GENILS (GENIUS, ILS e VND)

No trabalho de (Mine *et al.*, 2010) foi abordado o Problema de Roteamento de Veículos com Coleta e Entrega Simultânea (*Vehicle Routing Problem with Simultaneous Pickup and Delivery – VRPSPD*). Para resolvê-lo, foi proposto um algoritmo heurístico híbrido, denominado *GENILS*. Para gerar uma **solução inicial** foram utilizadas 3 heurísticas construtivas, adaptações do método da Inserção Mais Barata. A primeira foi a IMB-1R, a segunda foi IMB-NR, e a última foi uma heurística denominada VRGENIUS. A última possui duas fases, uma construtiva (VRGENI) e outra de refinamento (VRUS). Os procedimentos de ILS (*Iterated Local Search*) são combinados com a **busca local**, *Variable Neighborhood Descent (VND)*, que é utilizada para refinar a solução inicial. A *VND* explora a vizinhança de uma solução utilizando os movimentos *Shift*, *Shift(2,0)*, *Swap*, *Swap(2,1)*, *Swap(2,2)*, *M2-Opt* e *kOr-Opt*. Além disso, ela realiza uma intensificação da busca sempre que ocorre uma melhora na solução corrente. Essa intensificação é feita somente nas rotas modificadas e é realizada pelas buscas locais, pelos procedimentos *G3-opt* e *G4-opt* (os quais são baseados na heurística GENIUS) e pelo movimento *Reverse*. Para escapar dos ótimos locais utilizam-se 3 perturbações *múltiplos Shift*, *múltiplos swap*, *ejection chain*. A solução perturbada é refinada pela busca local, obtendo-se uma nova solução.

Em um conjunto consagrado de problemas-teste, foram alcançados todos os melhores resultados da literatura; em outro, foram gerados nove melhores resultados e seis resultados iguais aos melhores da literatura; em um terceiro, foram igualados três resultados da literatura, tendo-se uma diferença máxima igual a 3,16% para os demais problemas desse conjunto. Além disso, o GENILS obteve soluções com erro inferior a 1% em 67 dos 72 problemas teste, o que corresponde a 93% dos casos. De acordo com os resultados obtidos, verifica-se que o algoritmo proposto é competitivo com as melhores abordagens da literatura, sendo capaz de produzir soluções de qualidade.

3. CONSIDERAÇÕES FINAIS

A essa altura, é mister que se façam alguns comentários sobre as principais diferenças e semelhanças entre a dissertação ora apresentada e os trabalhos revisados anteriormente. O trabalho de dissertação ora apresentado utiliza uma abordagem híbrida para resolver um *VRPTW*. Na abordagem ora proposta, a **otimização global** é realizada pela meta-heurística de recozimento simulado, a mesma utilizada nos trabalhos de (Lin, Ying, Lee e Hsi, 2006), (Lin, Ying, Lee e Chen, 2006), e, (Sakakibara *et al.*, 2010). Para cada candidato gerado pela busca global, geram-se agrupamentos em função de janelas de tempo ou de capacidade. A **solução inicial** e as soluções subsequentes são geradas através de um novo operador de vizinhança, denominado de operador de vizinhança por quadrante. Podemos compará-lo ao método que (Chen *et al.*, 2015) gera sua solução inicial, utilizando uma ordenação de ângulos. Entretanto, esse novo método não utiliza somente os ângulos, mas, também o raio, a distância entre a escola e o depósito, subdividindo a área em quadrantes, que são percorridos por um sentido escolhido de forma aleatória. Para delimitar o tamanho dos agrupamentos de escolas, utilizaram-se como base as restrições de capacidade e tempo, acarretando um retorno ao depósito sempre que uma delas é violada. Trabalhos que utilizam pelo menos uma dessas restrições são (Lin, Ying, Lee e Hsi, 2006), (Lin, Ying, Lee e Chen, 2006), (Chen *et al.*, 2015). A **busca local** se preocupa em otimizar as rotas no interior de cada agrupamento, através da utilização do método de busca A* associado a uma heurística do passo seguinte.

A **função objetivo** é semelhante à utilizada no trabalho de (Pop e Chira, 2014), ou seja, é determinada pela soma das distâncias inter-agrupamentos com as distâncias intra-agrupamentos. Contudo, para o cálculo dessas distâncias, não se utiliza distâncias pré-calculadas entre pontos da rota e consultadas em tabelas, como no artigo revisado. No trabalho ora proposto, as mesmas são calculadas em tempo de execução, utilizando as distâncias percorridas em um mapa das ruas da região geográfica considerada.

Para avaliar as diversas contribuições desse trabalho, quatro outros métodos foram implementados. A descrição de cada um desses métodos pode ser vista a seguir.

O método 1, denominado de método GL realiza várias buscas globais. Após o término das mesmas, é realizada apenas um grupo de buscas locais para otimização da função objetivo. Esse método é denominado de global-local. A função objetivo foi denominada de função objetivo 2, e utiliza a distância euclidiana entre as escolas com apenas uma busca local, á semelhança do que ocorre no trabalho de (Guan *et al.*, 2010). Esse método utiliza o operador *swap* para geração de vizinhos. Esse operador realiza uma troca mútua entre duas escolas quaisquer de diferentes rotas. Essa operação de vizinhança já foi utilizada anteriormente em outros artigos como em (Lin, Ying, Lee e Chen, 2006), (Mine *et al.*, 2010) e (Sakakibara *et al.*, 2010), e, (He *et al.*, 2014).

O método 2, denominado de método GLG utiliza a abordagem global-local-g, ou seja, para cada busca global são realizadas g buscas locais para a função objetivo, essa função foi denominada de função objetivo 1. As buscas locais são realizadas através dos algoritmos A* ou Dijkstra, utilizando a distância de logradouro. Faz uso do método *swap* para geração de vizinhos.

O método 3, denominado de método QNG é utiliza uma busca global-local e utiliza o método de geração de vizinhos por quadrante, proposto nesse trabalho.

O método 4, denominado de método QNG_GLG utiliza a busca global-local-g e utiliza o método de geração de vizinhos por quadrante.

A comparação entre os métodos GL e GLG e entre QNG e QNG_GLG têm o mesmo objetivo, comparar a eficiência da abordagem global-local-g proposta nesse trabalho com a abordagem global-local. A comparação entre os métodos GL e QNG tem por objetivo verificar a eficiência do método de geração de vizinhos por quadrante, proposto como contribuição deste trabalho. Finalmente, a comparação do método proposto com o método

QNG_GLG tem a finalidade de avaliar a heurística do passo seguinte, proposta como uma contribuição deste trabalho. No capítulo de análise resultados analisaremos o resultado de todos os métodos com o resultado apresentado pelo método proposto.

A Tabela 4 mostra um resumo dos principais trabalhos analisados nessa revisão bibliográfica. Essa tabela possui as colunas ano, referindo-se ao ano de publicação do artigo; autor ou autores, referindo-se aos autores do artigo; título, que traz o título original do artigo, em inglês; problema/complexidade, que se refere ao problema resolvido pela abordagem do artigo e o grau de complexidade desse problema; base dados, que se refere aos dados que foram utilizados; método, que se refere ao método utilizado na abordagem apresentada; características, que se refere as características principais do problema, suas restrições, quantidade de depósitos e clientes, e o ambiente de desenvolvimento; medida de desempenho, que se refere a como foi medido o desempenho da abordagem apresentada; resultados, que apresenta os resultados obtidos por cada artigo.

Tabela 4 - Principais artigos abordados na revisão bibliográfica

Ano	Autor (es)	Título	Problema / Complexidade	Base de Dados	Método	Características	Medida de Desempenho	Resultados			
								Método	E5	E10	E14
2010	Sakakibara, K. Tsuda, T. Nishikawa, I.	<i>Simulated Annealing Method Based on Recursive Problem Decomposition for Vehicle Routing Problems</i>	PDP / NP-Hard	Criada pelo próprio autor	Método Híbrido (MH) de Recozimento Simulado (SA) com Técnica <i>Branch-and-Bound</i> (BB) em duas versões: LB1 com limite inferior calculado pela técnica de <i>Bound</i> , e LB2 limite inferior calculado pelo recozimento simulado.	Depósito: 1 Instâncias: 3 Nº de clientes: 5, 10, 14 Ambiente: Não informado	Menor valor da função objetivo / Tempo de execução	Método	E5	E10	E14
								MH (LB1)	236,76/ 1,1 s	305,17/ 27,7 s	362,14/ 176,9 s
								MH (LB2)	236,76/ 3,2 s	305,17/ 185,9 s	361,54/ 54640s
								SA	236,76/ 31,2 s	307,54/ 190,2 s	371,19/ 337,6 s
								BB	236,76/ 2,03 s	305,17/ 38918s	-
2010	Mine, M. T. Silva, M. S. A. Ochi, L. S. Souza, M. J. F. Silva, T. C. B.	<i>O problema de roteamento de veículos com coleta e entrega simultânea: uma abordagem via Iterated Local Search e GENIUS</i>	VRPSPD / NP-Hard/	benchmark do conjunto de problemas de Salhi e Nagy (1999), Dethloff (2001) e Montané e Galvão (2006)	<i>Algoritmo heurístico híbrido(Iterated Local Search e GENIUS)</i>	Depósito: 1 Nº de veículos: variável com capacidades iguais Nº de clientes: variável com demandas de coleta e entrega. Ambiente: Microsoft Visual C++	Quantidade de acertos do total de problemas, e gap máximo em relação a melhor solução apresentada na literatura	benchmarks	Qtde de acertos/total	Gap máximo	
								Salhi e Nagy	4/14	3,16%	
								Dethloff	40/40	0%	
								Montané e Galvão	15/18	0,58%	

2012	Jinhui, G. Ying, X. Hongzhen, W.	<i>An Improved TS for the Open Vehicle Routing Problem with Soft Time Windows</i>	<i>OVRPSTW / NP-Hard/</i>	benchmark do conjunto de dados dos problemas de (Solomon, 1987), os conjuntos de dados independentes utilizados foram R1, R2, RC1, RC2	Melhoria na heurística Busca Tabu (TS)	Depósito: 1 (Sem necessidade de retorno ao depósito) Nº de veículos: variável Nº de clientes: variável de 100 a 200 Ambiente: JAVA	Taxa de melhoria da TS apresentada no trabalho em comparação com os trabalhos de Repoussis e Fenghua - nº de veículos/ custo	Dados				Repoussis	Fenghua
								R1	37,60% / 18,03%		15,33% / 11,33%		
								R2	35,28% / 14,44%		18,3% / 1,16%		
								RC1	27,3% / 13,88%		15,33% / 5,7%		
								RC2	42,86% / 26,27%		33,33% / 15,88%		
2014	Pop, P. Chira, C.	<i>A Hybrid Approach based on Genetic Algorithms for Solving the Clustered Vehicle Routing Problem</i>	<i>CluVRP / NP-Hard</i>	20 instâncias CVRP descritas por Golden <i>et al.</i>	Conexão Global/ Conexão Local/ Algoritmo Genético (GA)	Depósito: 1 Nº clusters: variável de 1 a 71 Nº de veículos: variável de 17 a 97 Nº de clientes: 241, 321, 401, 481 Ambiente: Java, JDK 1.6	Comparação da média das soluções encontradas (AFS) com a melhor solução (BS), em termos de distância percorrida	Nº cliente	Nº cluster	BS	AFS		
								241	49	5946	6066		
								321	65	9345	9574		
								401	81	12471	12898		
								481	97	16498	16977		
2015	Chen, M. H. Chago, P. C. Chiu, C. Y. Annadurai, S. P.	<i>A Hybrid Two-stage Sweep Algorithm for Capacitated Vehicle Routing Problem</i>	<i>CVRP / NP-Hard</i>	Benchmark de problemas obtidos no Website internacional VRP (http://neo.lcc.um.a.es/radi-aeb/WebVRP/)	Algoritmos de varredura / Busca Gulosa / Recombinação de clusters	Depósito: 1 Instâncias: 10 Nº de veículos: 3,4,5,7,8,10,14,19 Nº de clientes: 23,30,33,51,76,101,484 Ambiente: Não informado	Comparação com o BRA através da taxa de erro em relação a solução ótima	A média das taxas de erro obtidas pela abordagem proposta foi de 3,32% e a média das taxas de erro pelo uso do algoritmo BRA foi de 7,50%					

FUNDAMENTAÇÃO TEÓRICA

O capítulo anterior apresentou a pesquisa bibliográfica realizada que norteia esta dissertação. A mesma relatou as fontes utilizadas, as publicações com algumas derivações do problema de roteamento de veículos, e as principais abordagens desenvolvidas no decorrer do tempo para a resolução desse problema. Finaliza-se o capítulo anterior com uma comparação entre alguns dos trabalhos revisados e a dissertação ora apresentada. Neste capítulo será abordada de forma breve a fundamentação teórica que subsidiará o desenvolvimento do trabalho de dissertação, quais sejam: grafos (definição, incidência, adjacência, grau, caminho, comprimento, subcaminho, circuito, ciclo, laço, grafo conexo, componentes conexos, grafo fortemente conexo, componentes fortemente conexos, grafo valorado, e, representações de grafos) e estratégias de busca sem informação.

3.1 GRAFOS

Muitas situações do mundo real podem ser convenientemente descritas por um diagrama que consiste de um conjunto de vértices, sendo certos pares de vértices unidos através de arestas. Por exemplo, os vértices podem representar centros de comunicação, com linhas que representam links de comunicação. No presente trabalho, os vértices representam cruzamentos de logradouros e as arestas representam as ruas e avenidas.

3.1.1 Definição

Um grafo é um par $G = (V, E)$, em que o conjunto de arestas E consiste de pares de vértices não ordenados. Por exemplo, uma aresta é definida por $\{u, v\}$, em que $u, v \in V$ e $u \neq v$. Por convenção, usa-se a notação (u, v) para uma aresta, ao invés da notação $\{u, v\}$, e

considera-se (u, v) e (v, u) como sendo a mesma aresta. A Figura 8(a) é uma representação de um grafo não direcionado com o conjunto de vértices $\{1,2,3,4,5,6\}$ (Cormen, 2009).

3.1.2 Dígrafo ou grafo direcionado

Um dígrafo ou grafo direcionado, G é um par (V, E) , onde V é um conjunto finito e E é uma relação binária em V . O conjunto V é chamado de conjunto dos vértices de G , e os elementos desse são chamados vértices. O conjunto E é chamado conjunto de **arcos** de G . A Figura 8(b) é uma representação de um grafo direcionado, onde o conjunto de vértices é $\{1,2,3,4,5,6\}$. Os vértices são representados por círculos na figura, e os arcos são representados pelas setas. Diferentemente de uma aresta, os arcos são orientados. Assim, um $(1,2)$ é diferente de um arco $(2,1)$. Pode-se notar que é possível que um vértice tenha uma seta que se inicia e termina nele mesmo referindo-se ao conceito de **laço** (Cormen, 2009).

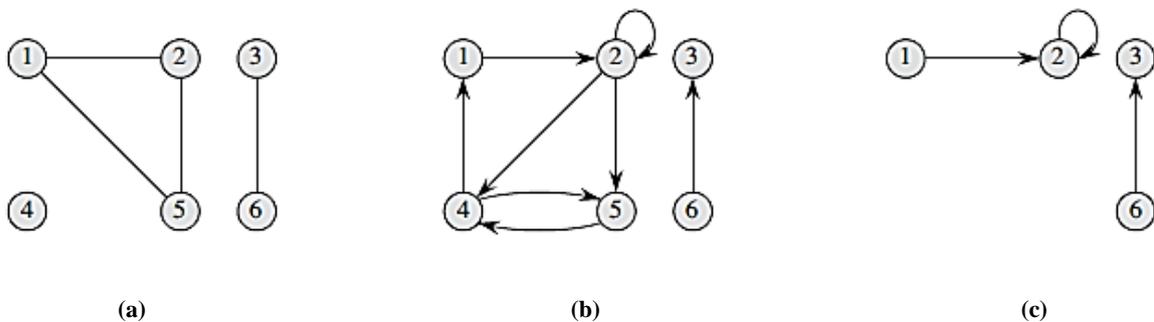


Figura 8 – Grafo e dígrafo. (a) Um grafo não direcionado $G = (V, E)$, onde $V = \{1, 2, 3, 4, 5, 6\}$ e $E = \{(1, 2), (1, 5), (2, 5), (3, 6)\}$. O vértice 4 é isolado. (b) Um dígrafo $G = (V, E)$, onde $V = \{1, 2, 3, 4, 5, 6\}$ e $E = \{(1, 2), (2, 2), (2, 4), (2, 5), (4, 1), (4, 5), (5, 4), (6, 3)\}$. O arco $(2, 2)$ é um laço ou *auto-loop*. (c) O subgrafo do grafo da parte (a) inferido pelo conjunto de vértices $\{1, 2, 3, 6\}$.

Fonte: (Cormen, 2009)

A Figura 8 será utilizada como base para o enunciado de alguns conceitos básicos e importantes sobre grafo.

3.1.3 Incidência

Se (u, v) é um arco de um grafo direcionado $G(V, E)$, diz-se que (u, v) é incidente do, ou parte do vértice u e é incidente para, ou entra no, vértice v . Por exemplo, os arcos que

partem do vértice 2 na Figura 8(b) são (2,2), (2,4) e (2,5). Os arcos que entram no vértice 2 são (1,2) e (2,2). Se (u, v) é uma aresta de um grafo não direcionado $G(V, E)$, diz-se que (u, v) é incidente nos vértices u e v . Na Figura 8(a), as arestas incidentes no vértice 2 são (1,2) e (2,5).

3.1.4 Adjacência

Se (u, v) é um arco de um grafo $G(V, E)$, diz-se que o vértice v é adjacente (ou **vizinho**) ao vértice u . Quando o grafo é não direcionado, a relação de adjacência é simétrica. Quando um grafo é direcionado, a relação de adjacência não é necessariamente simétrica. Se v é adjacente a u em um grafo direcionado, pode-se escrever $u \rightarrow v$. Nas partes (a) e (b) da Figura 8, o vértice 2 é adjacente do vértice 1, visto que a aresta (1,2) pertence a ambos os grafos. O vértice 1 não é adjacente do vértice 2 na Figura 8(b), visto que o arco (2,1) não pertence ao grafo. Quando todos os seus vértices são adjacentes o grafo é **completo**.

3.1.5 Grau

O grau de um vértice em um grafo não direcionado é o número de arestas incidentes nesse vértice. Por exemplo, o vértice 2 na Figura 8(a) tem grau 2. Um vértice cujo grau é 0, tal como o vértice 4 na Figura 8(a) é isolado. Em um grafo direcionado, o grau de saída de um vértice é o número de arcos que partem deste (divergente), e o grau de entrada de um vértice é o número de arcos que entram neste (convergente). O grau de um vértice em um grafo direcionado é a soma do grau de entrada e saída. O vértice 2 na Figura 8(b) tem grau de entrada 2, grau de saída 3 e grau 5. Se todos os vértices de um grafo tiverem o mesmo grau ele é chamado de **grafo regular**.

3.1.6 Caminho e comprimento

O caminho de tamanho k do vértice u para o vértice u' em um grafo $G = (V, E)$ é uma sequência $(v_0, v_1, v_2, \dots, v_k)$ de vértices tal que $u = v_0$, $u' = v_k$ e $(v_{i-1}, v_i) \in E$ para $i = 1, 2, \dots, k$. O **comprimento do caminho** é o número de arestas do caminho. O **caminho** é formado pelos vértices $v_0, v_1, v_2, \dots, v_k$ e as arestas $(v_0, v_1), (v_1, v_2), \dots, (v_{k-1}, v_k)$. (Existe sempre um caminho de comprimento 0 de u para u). Se existe um caminho p de u para u' , diz-se que u' é **alcançável** de u via p . Um caminho é **simples** se todos os vértices no caminho são distintos. Na Figura 8(b), o caminho $(1,2,5,4)$ é um caminho simples de comprimento 3. O caminho $(2,5,4,5)$ não é simples.

Existem dois tipos de caminho que são conhecidos no âmbito dos grafos: o **caminho Hamiltoniano** que é aquele que contém cada vértice do grafo exatamente uma vez, e o **caminho Euleriano** que é aquele que contém cada aresta do grafo exatamente uma vez.

3.1.7 Subcaminho

Um **subcaminho** de um caminho $p = (v_0, v_1, v_2, \dots, v_k)$ é uma sequência contínua de vértices. Tal que, para qualquer $0 \leq i \leq j \leq k$, a subsequência de vértices $(v_i, v_{i+1}, \dots, v_j)$ é um subcaminho de p .

3.1.8 Circuito, ciclo, e, laço

Em um grafo direcionado, um caminho $(v_0, v_1, v_2, \dots, v_k)$ forma um **circuito** se $v_0 = v_k$ e o caminho contém pelo menos um **arco**. Um **ciclo** é um circuito onde todos os vértices v_1, v_2, \dots, v_{k-1} são distintos. Um **auto-loop**, ou **laço**, é um circuito de comprimento 1. Dois caminhos $(v_0, v_1, v_2, \dots, v_{k-1}, v_0)$ e $(v'_0, v'_1, v'_2, \dots, v'_{k-1}, v'_0)$ formam o mesmo circuito se existe um inteiro j tal que $v'_i = v_{(i+j) \bmod k}$ para $i = 0, 1, \dots, k-1$. Na Figura 8(b), o caminho $(1,2,4,1)$ forma o mesmo circuito que os caminhos $(2,4,1,2)$ e $(4,1,2,4)$. Esse

circuito descrito é um ciclo, mas o circuito (1,2,4,5,4,1) não é. O circuito (2,2) formado pelo arco (2,2) é um laço. Um grafo direcionado sem laços é simples. Em um grafo não direcionado, o caminho $(v_0, v_1, v_2, \dots, v_k)$ forma um circuito se $k \geq 3$ e $v_0 = v_k$. Um grafo sem ciclos é **acíclico**. Um **ciclo Hamiltoniano** acontece quando o caminho for um caminho Hamiltoniano. Se um grafo contiver um ciclo Hamiltoniano, então ele é considerado um **grafo Hamiltoniano**. Se o grafo $G(V, E)$ possui um circuito que contém todas as suas arestas esse é um **grafo Euleriano**.

3.1.9 Grafo conexo, e componentes conexos

Um grafo não direcionado é **conexo** se todos os vértices são alcançáveis de todos para todos. Os componentes conexos de um grafo são as classes de equivalência de vértices no âmbito da relação “é alcançável de”. O grafo da Figura 8(a) tem três componentes conexos: $\{1,2,5\}$, $\{3,6\}$ e $\{4\}$. Todos os vértices contidos em $\{1,2,5\}$ são alcançáveis de todos os outros vértices contidos em $\{1,2,5\}$.

3.1.10 Grafo fortemente conexo, e componentes fortemente conexos

Um grafo direcionado é fortemente conexo se cada dois vértices desse são alcançáveis entre si. Os componentes fortemente conexos de um grafo direcionado são as classes de equivalência de vértices no âmbito da relação “são mutuamente alcançáveis”. O grafo na Figura 8(b) tem três componentes fortemente conexos: $\{1,2,4,5\}$, $\{3\}$, e $\{6\}$. Todos os pares de vértices contidos em $\{1,2,4,5\}$ são mutuamente alcançáveis. Os vértices $\{3,6\}$ não formam um componente extremamente conexo, pois o vértice 6 não pode ser alcançado do vértice 3.

3.1.11 Grafo valorado

Um grafo valorado é um grafo em que cada aresta tem um valor associado. Formalmente, um grafo valorado $G = (V, E)$ consiste de um conjunto $V = \{v_1, v_2, \dots, v_k\}$ de

vértices, e um conjunto $E = \{(v_1, v_2, P_{v_1 \rightarrow v_2}), \dots, (v_{k-1}, v_k, P_{v_{k-1} \rightarrow v_k})\}$ de arestas e pesos, onde $P_{v_{k-1} \rightarrow v_k}$ representa o valor (peso) associado à aresta que conecta os vértices v_{k-1} e v_k pertencentes a V .

3.1.12 Representações de grafos

Existem duas formas padrões de se representar um grafo $G(V, E)$: como uma coleção de listas de adjacência ou como uma matriz de adjacências. Através dessas duas formas consegue-se representar tanto grafos não direcionados como os direcionados. No entanto, a lista de adjacências é mais utilizada quando se trata de espaço de armazenamento, por armazenar somente o conjunto das arestas $|E|$, e esse ser muito menor que $|V|^2$ utilizado pela matriz de adjacências. A forma de representação por matriz de adjacências é mais utilizada quando se precisa de um tempo relativamente rápido de pesquisa para responder se dois vértices estão conectados por uma aresta ou não. Na Figura 9, são vistas as duas formas de representação aplicadas a um grafo não direcionado.

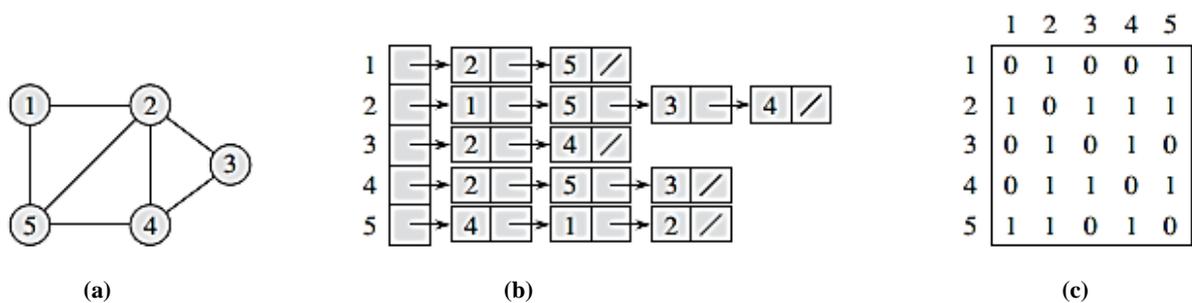


Figura 9 – Grafo não direcionado $G = (V, E)$, onde $V = \{1, 2, 3, 4, 5\}$ e $E = \{(1, 2), (2, 1), (1, 5), (5, 1), (2, 5), (5, 2), (2, 4), (4, 2), (2, 3), (3, 2), (3, 4), (4, 3), (4, 5), (5, 4)\}$. (a) representação gráfica. (b) representação por lista de adjacências. (c) representação por matriz de adjacências.
 Fonte: (Cormen, 2009)

Na Figura 10, mostra-se as formas de representação de um dígrafo.

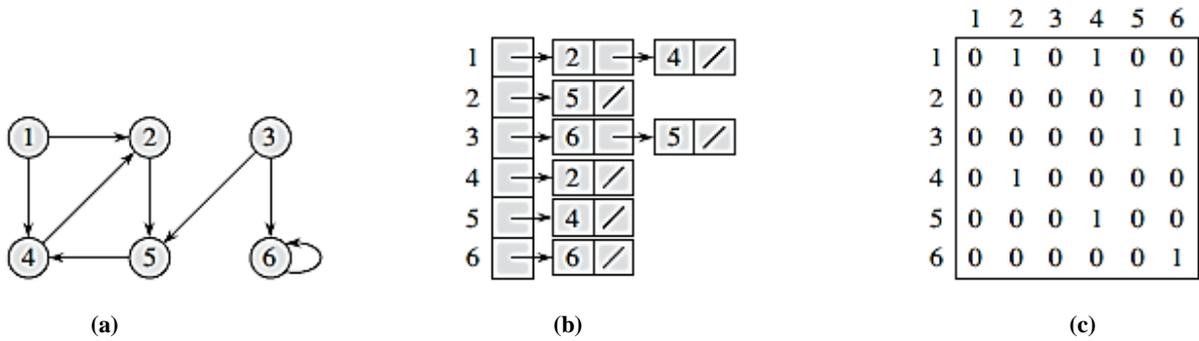


Figura 10 - Grafo direcionado $G = (V, E)$, onde $V = \{1, 2, 3, 4, 5, 6\}$ e $E = \{(1, 2), (1, 4), (2, 5), (3, 5), (3, 6), (4, 2), (5, 4), (6, 6)\}$. O arco $(6, 6)$ é um laço ou auto-loop. (a) representação gráfica. (b) representação por lista de adjacências. (c) representação por matriz de adjacências. Fonte: (Cormen, 2009)

A Figura 11 mostra as representações usuais de um grafo valorado:

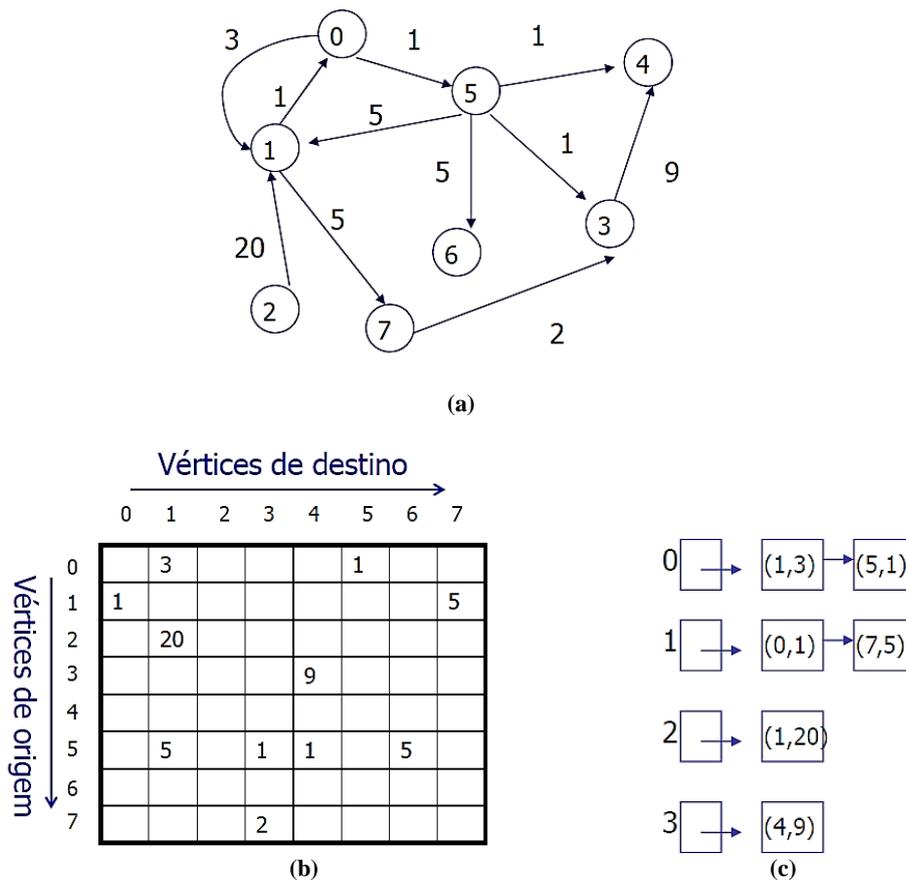


Figura 11 – Grafo dirigido e valorado $G(V, E, P)$. (a) representação gráfica. (b) representação em matriz de adjacências (ou matriz de valores), vértices de origem são ligados aos vértices de destino com os pesos atribuídos na célula que os conecta. (c) representação em lista de adjacências, um exemplo seria o vértice 0 é conectada ao vértice 1 com peso 3 e ao vértice 5 com peso 1.

Os grafos são utilizados em nosso trabalho para representar o mapa da região IV, onde, os encontros de logradouros foram considerados os vértices, e, os próprios logradouros as arestas. Trata-se de um grafo dirigido, ou dígrafo, valorado e fortemente conexo. Algumas ruas sem saída foram consideradas como laços e foram tratadas para não causar problemas na busca por menores caminhos.

3.2 AGENTES DE RESOLUÇÃO DE PROBLEMAS

Processos de busca são agentes que recebem um problema como entrada e retornam uma solução sob a forma de uma sequência de ações (Russel e Norvig, 2004).

Um problema pode ser definido formalmente por quatro componentes: estado inicial, função sucessor, teste de objetivo e custo de caminho. Suas descrições podem ser vistas na Tabela 5.

Através de simulações, utilizando as possíveis ações em cada estado, os agentes podem construir diversos percursos hipotéticos, do estado atual ao objetivo desejado, escolhendo aquele que resulta em um melhor valor para uma medida de desempenho que mensura o custo do caminho.

Tabela 5 – Componentes da definição formal de um problema

ESTADO INICIAL	O estado inicial que o agente começa.
FUNÇÃO SUCESSOR	Uma Descrição de ações possíveis que estão disponíveis para o agente.
TESTE DE OBJETIVO	Determina se um dado estado é um estado objetivo.
CUSTO DE CAMINHO	Atribui um custo numérico a cada caminho.

Outros conceitos pertinentes a um processo de busca são:

- Espaço de estados: Junção do espaço inicial e da função sucessora, organizando o conjunto de todos os espaços acessíveis a partir do estado inicial.
- Caminho: Uma seqüência de estados conectados por uma seqüência de ações.
- Custo de passo: É o custo de adotar a ação a para ir do estado x ao estado y é denotado por $c(x,a,y)$.
- Solução: é um caminho desde o estado inicial até o estado objetivo.
- Solução ótima: é a solução com o menor custo de caminho entre todas as soluções.

Existem dois tipos de buscas: buscas sem informação e buscas com informação. A Figura 12, adaptado de (Souza, 2008), mostra as subdivisões dos algoritmos de busca.

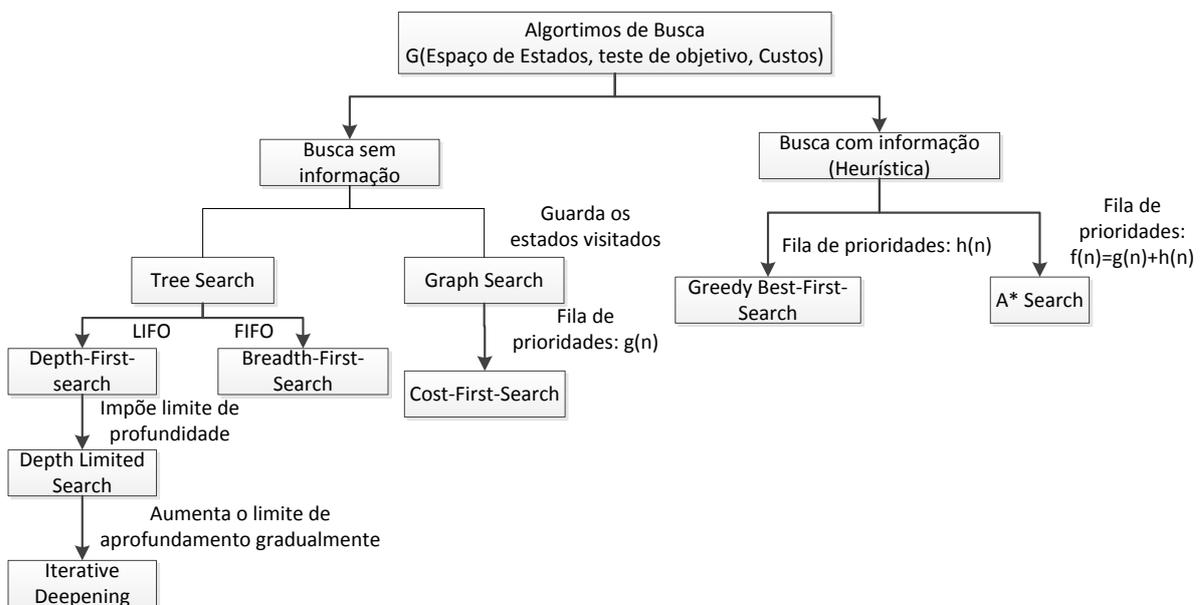


Figura 12 – Algoritmos de busca e suas subdivisões adaptado de (Souza, 2008)

3.3 BUSCA SEM INFORMAÇÃO (BUSCA CEGA)

A busca sem informação, conhecida como busca cega, não dispõe de nenhuma informação adicional sobre estados, a não ser aquelas fornecidas na definição do problema. Tudo que conseguem fazer é gerar sucessores e distinguir estados objetivos dos não-objetivos. As estratégias desse tipo de busca são definidas em função da ordem em que os nodos são expandidos (Russel e Norvig, 2004).

3.3.1 Busca em largura (*Breadth-First-Search – BFS*)

A busca em extensão é uma estratégia simples em que o nodo raiz é expandido primeiro. Em seguida todos os sucessores do nodo raiz são expandidos. Depois os sucessores desses nodos e assim por diante. Todos os nodos em uma dada profundidade da árvore de busca são expandidos, antes que todos os nodos do nível seguinte sejam expandidos (Russel e Norvig, 2004).

A busca pode ser implementada como uma fila do tipo first-in-first-out (FIFO), ou seja, os nodos visitados primeiro serão expandidos primeiro. A fila FIFO coloca todos os sucessores gerados no final da fila. Os nodos com baixa profundidade serão expandidos antes dos nodos mais profundos. Tecnicamente, a busca em extensão eventualmente encontrará o nodo ótimo após expandir todos os nodos mais rasos. Essa estratégia nem sempre é a preferida, considerando a quantidade de tempo e memória que é empregada para se completar uma busca. Todo o nodo gerado deve permanecer na memória, porque faz parte da borda ou é um ancestral de um nodo de borda.

Os requisitos de memória são um problema maior do que o tempo de execução. Um problema com fator de ramificação $b=10$, e de profundidade 8, exigiria um computador com uma memória da ordem de terabytes. Em se tratando de tempo de execução, considerando um problema com o mesmo fator de ramificação, uma solução de profundidade 12 requereria um total de 35 anos para que a busca em extensão a alcançasse (Russel e Norvig, 2004).

3.3.2 Busca em profundidade (*Depth-First-Search – DFS*)

A busca em profundidade sempre expande o nodo mais profundo na borda atual da árvore de busca. Ela prossegue imediatamente até o nível mais profundo da árvore, onde os nodos não possuem sucessores. Na medida em que os nodos são expandidos, eles são retirados da borda. Então a busca retorna ao nodo seguinte mais raso que ainda possui sucessores

inexplorados. Essa estratégia pode ser implementada com uma fila last-in-first-out (LIFO), também conhecida como pilha, onde o último elemento gerado é o primeiro a ser explorado.

No que diz respeito a requisitos de memória, essa busca só precisa armazenar um único caminho da raiz até um nó de folha, juntamente com os nós restantes não expandidos de cada nó no caminho. Uma vez que o nó é expandido, ele pode ser removido da memória, se todos os seus descendentes já tiverem sido explorados. Em comparação com a busca em extensão, levando em consideração um fator de ramificação $b=10$ uma busca em profundidade 12 ocupa um espaço 10 bilhões de vezes menor (Russel e Norvig, 2004).

A desvantagem da busca em profundidade é que, ao fazer a escolha de um ramo onde não se encontra a solução, pode ficar paralisada ao descer um caminho muito longo. Consequentemente, a busca em profundidade não é ótima. Se uma subárvore tiver uma profundidade ilimitada, mas não contivesse nenhuma solução, a busca em profundidade nunca terminaria (Russel e Norvig, 2004).

3.3.3 Dijkstra (*Cost-First-Search* – *CFS*)

O algoritmo de Dijkstra foi desenvolvido por (Dijkstra, 1959). É também conhecido como algoritmo de busca de custo uniforme (*Cost-First-Search*). Esse algoritmo calcula o custo mínimo entre vértices de um grafo. A partir do vértice atual, o algoritmo expande o vértice vizinho com menor custo de caminho até o momento. O custo associado a cada vértice n vizinho a ser escolhido é medido pela função $g(n)$, que é o custo do caminho da origem até o vértice n . Caso a expansão não dependesse do custo dos vizinhos, ou seja, se os custos fossem todos iguais o algoritmo ficaria reduzido a uma busca por extensão. A seguir, na Figura 13, podemos ver o pseudo código do algoritmo de Dijkstra.

```

para todo  $v \in V[G]$  //v são o vértices do conjunto de vértices do grafo G
     $d[v] \leftarrow \infty$  //estimativa de custo mínimo do vértice v
     $\pi[v] \leftarrow -1$  //precedente de v
 $d[s] \leftarrow 0$  //estimativa de custo mínimo do vértice raiz

 $Q \leftarrow V[G]$  //conjunto dos vértices que ainda não possuem estimativa mínima

enquanto  $Q \neq \emptyset$  //enquanto o conjunto de vértices não é vazio
     $u \leftarrow \text{extrair-mín}(Q)$  //extrai o vértice u de Q,  $Q \leftarrow Q - \{u\}$ 
    para cada v adjacente a u //procura os vértices adjacentes a u
        se  $d[v] > d[u] + w(u, v)$  //relaxa (u, v)
            então  $d[v] \leftarrow d[u] + w(u, v)$  //custo de v é modificado
                 $\pi[v] \leftarrow u$  //u é atribuído como precedente de v

```

Figura 13 – Pseudocódigo do algoritmo Dijkstra

O algoritmo de Dijkstra aplicado neste trabalho possui uma particularidade. Por se utilizar o algoritmo várias vezes no cálculo da função objetivo, não foi possível percorrer o espaço de vértices total. A aplicação restringiu-se a encontrar um caminho do vértice inicial até o vértice objetivo. Quando o vértice objetivo é encontrado apresenta-se a solução. Por esse motivo o algoritmo funciona de maneira gulosa, mas nem sempre encontra o menor caminho entre os vértices de início e objetivo. Esse fato ocorre pelo algoritmo ser utilizado varias vezes separadamente de escola para escola, de modo que o menor caminho é uma junção dos menores caminhos, essa junção nem sempre encontra o menor caminho entre os nodos. O capítulo de resultados exemplificará o que ora afirmamos.

3.4 BUSCA COM INFORMAÇÃO (HEURÍSTICA)

As estratégias de busca com informação são aquelas que utilizam o conhecimento específico do problema, podendo as mesmas encontrar soluções de modo mais eficiente que uma estratégia sem informação. A abordagem geral é conhecida como busca pela melhor escolha. A busca pela melhor escolha é uma especialização do algoritmo geral de busca em árvore ou grafo, na qual um nodo é selecionado para expansão com base em uma função de avaliação $f(n)$. Quando a função tem que ser minimizada, o nodo com a avaliação mais baixa é selecionado para expansão (Russel e Norvig, 2004).

3.4.1 Função Heurística

Um componente fundamental desses algoritmos é uma função heurística, denotada por $h(n)$:

$h(n)$ = custo estimado do caminho mais econômico do nodo n até um nodo objetivo

As funções heurísticas são a forma mais comum de aplicar conhecimento adicional do problema ao algoritmo de busca. Porém, se quisermos descobrir as soluções mais curtas usando, por exemplo, o algoritmo A^* , precisa-se de uma função heurística que não superestime o número de passos até o objetivo.

Uma maneira de caracterizar a qualidade de uma heurística é o fator de ramificação efetiva, b^* . Se o número total de nodos gerados por A^* para um determinado problema é N , e se a profundidade da solução é d , então b^* é o fator de ramificação que uma árvore uniforme de profundidade d precisa ter para conter $N + 1$ nodos. Desse modo:

$$N + 1 = 1 + b^* + (b^*)^2 + \dots + (b^*)^d \quad (1)$$

O fator de ramificação efetiva pode variar em várias instâncias do problema. Em geral, ele é relativamente constante para problemas suficientemente difíceis. Uma heurística bem projetada teria um valor b^* próximo de 1, permitindo a resolução de problemas bastante extensos.

Um problema com menos restrições sobre as ações é chamado um problema relaxado. O custo de uma solução ótima para um problema relaxado é considerado uma heurística admissível para o problema original. A heurística é admissível, pois a solução ótima no problema original, também é uma solução no problema relaxado.

Heurísticas como a distância de Manhattan e a distância Euclidiana são consideradas heurísticas admissíveis. Sendo dois pontos no espaço euclidiano, P_1 com coordenadas (x_1, y_1)

e P_2 com coordenadas (x_2, y_2) a distância de Manhattan entre eles será $|x_1 - x_2| + |y_1 - y_2|$.

A distância em linha reta, ou Euclidiana será $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

3.4.2 Busca gulosa pela melhor escolha (*Greedy Best-First-Search – GBFS*)

A busca gulosa pela melhor escolha, tenta expandir o nodo mais próximo à meta, na suposição de que isso provavelmente levará a uma solução rápida. Desse modo a avaliação de nodos usa apenas a função heurística: $f(n) = h(n)$.

A solução através do algoritmo de busca gulosa, com uma heurística de distância em linha reta, encontra uma solução sem expandir nenhum nodo que não esteja no caminho da solução. Consequentemente, seu custo de busca é mínimo. Porém, ela não é ótima, pois sempre escolhe expandir os nodos pelo mais próximo. Ela é incompleta, pode entrar em um caminho infinito e nunca retornar para testar outras possibilidades. Um exemplo de busca gulosa para encontrar o caminho de A até J pode ser visto na Figura 14. Para cada nodo, a função heurística h é o valor de sua distância em linha reta até o nodo objetivo J.

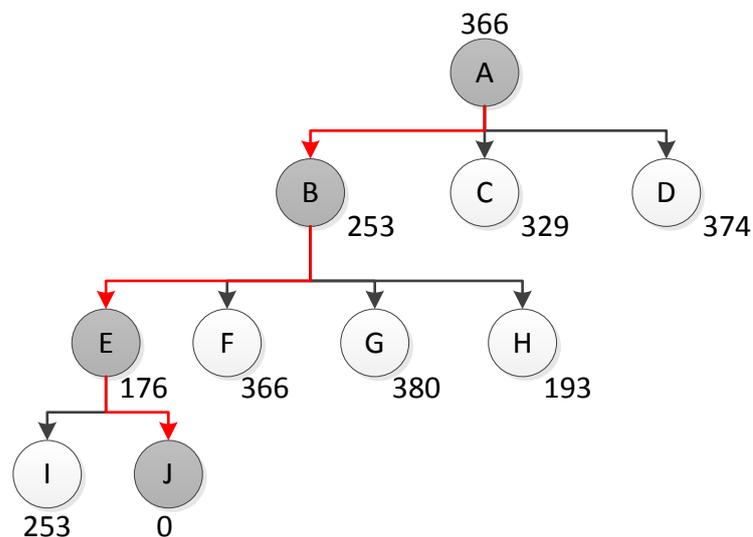


Figura 14 – Exemplo de uma busca gulosa com a função heurística: $f(n)=h(n)$.

3.4.3 Busca A estrela (*A Star Search* – A^*)

A forma mais conhecida e que resulta em uma melhor escolha, é a busca A^* . Ela avalia nodos combinando $g(n)$, o custo para alcançar um dado nodo, e $h(n)$, o custo para ir desse nodo até o nodo objetivo. Isso é mostrado na equação (2):

$$f(n) = g(n) + h(n) \quad (2)$$

Tendo em vista que $g(n)$ fornece o custo de caminho desde o nodo inicial até o nodo n , e que $h(n)$ é o custo estimado do caminho de custo mais baixo desde n até o objetivo, temos que:

$$f(n) = \text{custo estimado da solução de custo mais baixo passando por } n$$

Uma das opções para se encontrar a solução de custo mais baixo é experimentar primeiro o nodo com o menor valor de $g(n) + h(n)$. Essa estratégia passa a ser mais que razoável, desde que a função heurística $h(n)$ satisfaça a certas condições. A busca A^* será ao mesmo tempo completa e ótima (Russel e Norvig, 2004).

A análise do caráter ótimo de A^* é direta se usada com uma busca em árvore. Nesse caso, A^* será ótima se $h(n)$ for uma heurística admissível – isto é, desde que $h(n)$ nunca superestime o custo para alcançar o objetivo. Heurísticas admissíveis sempre acreditam que o custo da resolução do problema é menor do que ela realmente é.

Como visto na abordagem anterior das funções heurísticas, um bom exemplo de heurística seria a distância em linha reta. O caminho mais curto entre dois pontos quaisquer será sempre uma linha reta. Assim a linha reta nunca pode ser uma superestimativa.

Se a busca A^* for usada com um algoritmo de busca em grafo, soluções não ótimas podem ser retornadas. A busca em grafo pode descartar o caminho ótimo para um estado repetido, se ele não for o primeiro caminho gerado. Há duas maneiras de corrigir esse

problema. A primeira solução é estender a busca em grafo de tal forma que a mesma descarte o caminho mais dispendioso entre dois caminhos quaisquer descobertos para o mesmo nodo. A segunda é assegurar que o caminho ótimo para qualquer estado repetido seja sempre o primeiro a ser seguido. Essa propriedade será válida se impusermos um requisito extra sobre $h(n)$, ou seja, o requisito de consistência (monotonicidade). Uma heurística $h(n)$ é consistente se, para todo nodo n e todo sucessor n' de n gerado por qualquer ação a , o custo estimado de alcançar o objetivo a partir de n não é maior que o custo do passo de se chegar a n' somado ao custo estimado de alcançar o objetivo a partir de n' . Na Figura 15 é mostrado um fluxo básico da busca A*. Um exemplo de busca A* pode ser visto na Figura 16, onde o algoritmo parte do nodo A com a finalidade de encontrar o nodo J usando o menor caminho. Comparado com a busca gulosa no exemplo apresentado das cidades A até a cidade J em (Russel e Norvig, 2004), o A* consegue uma melhoria de 32 km em relação ao primeiro.

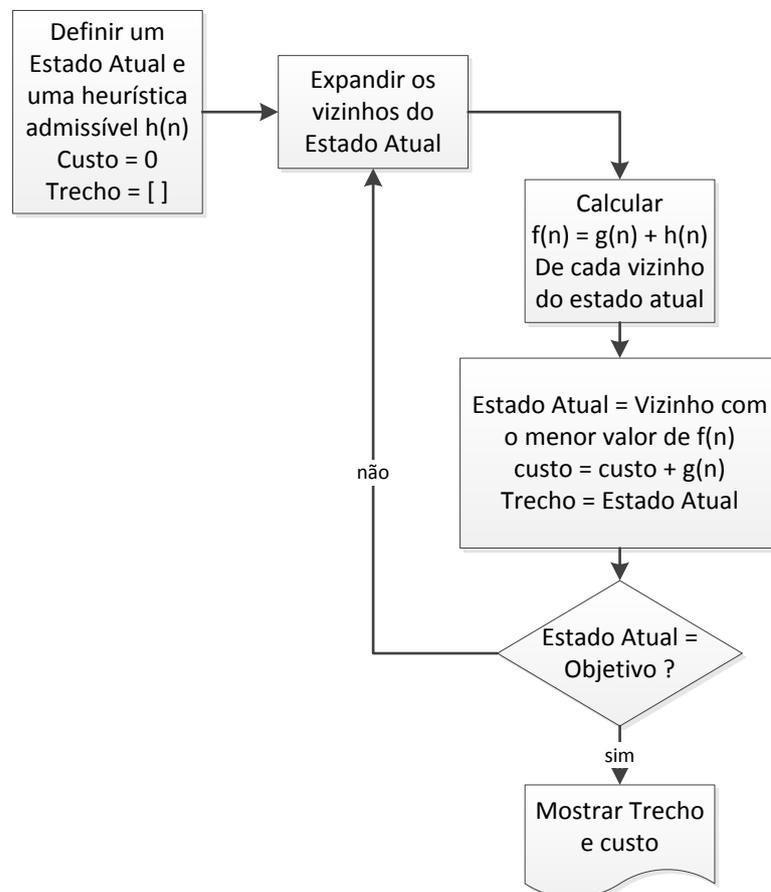


Figura 15 – Fluxograma básico do algoritmo A estrela.

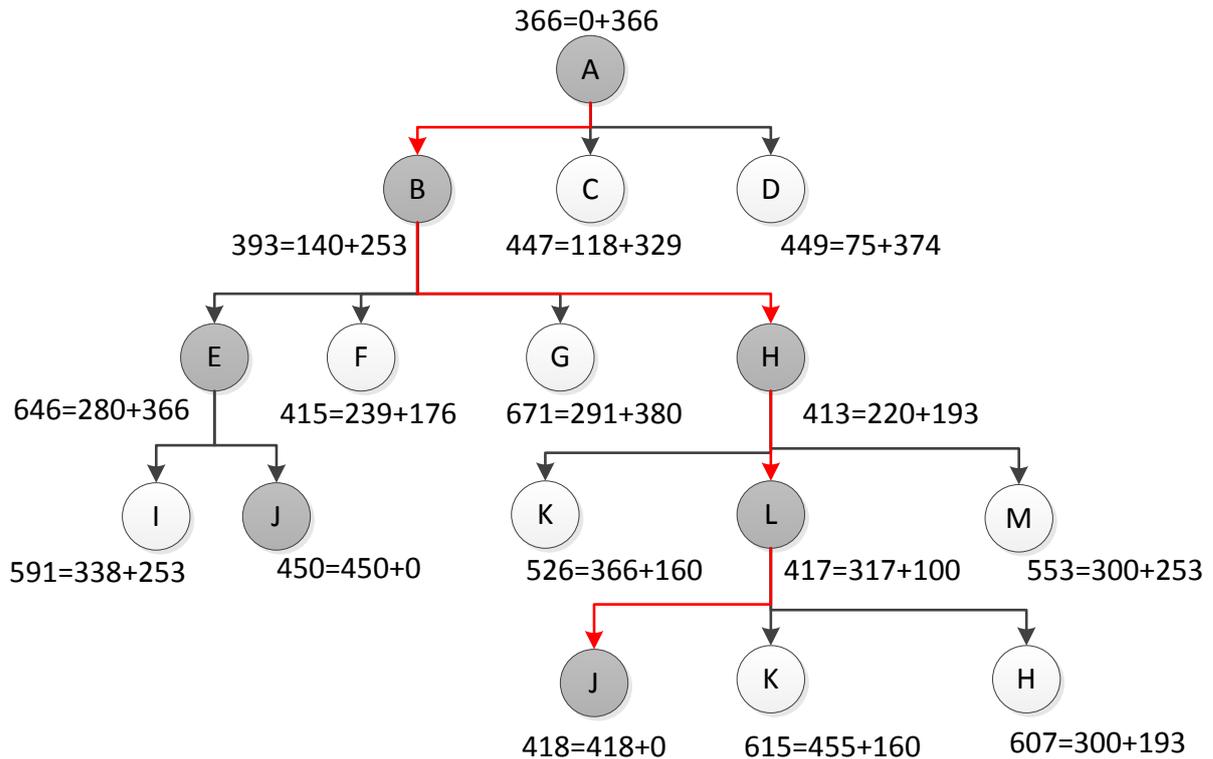


Figura 16 – Exemplo de uma busca A estrela com função heurística $f=g+h$.

3.5 ALGORITMO HEURÍSTICO GLOBAL

Algoritmos heurísticos globais, ou de busca global, encontram o melhor caminho através da otimização de uma função objetivo. Dependendo do problema, procura-se minimizar ou maximizar o valor da função objetivo. Nesse trabalho, na procura da melhor rota, associamos um algoritmo de busca global, recozimento simulado, ao algoritmo A*, aplicado a buscas locais, dentro de agrupamentos. A seguir descrevemos esse algoritmo de busca global.

3.5.1 Reozimento Simulado (*Simulated Annealing - SA*)

O método de recozimento simulado é baseado em um processo físico, o recozimento térmico de sólidos aquecidos criticamente. Quando um sólido é trazido a um estado fundido por aquecimento a uma temperatura elevada, os átomos do metal fundido movem-se livremente em relação uns aos outros. No entanto, os movimentos de átomos se restringem a

medida que a temperatura é reduzida. Com a diminuição gradativa e lenta da temperatura, os átomos tendem a assumir padrões bem definidos, atingindo formas cristalinas, que apresentam uma energia potencial mínima. O processo de formação de cristais depende essencialmente da velocidade de arrefecimento. Quando a temperatura do metal fundido é reduzida a uma velocidade muito rápida, o mesmo pode não ser capaz de atingir o estado cristalino e sim um estado policristalino, com energia potencial maior do que a do estado cristalino. Em aplicações de microeletrônica, na obtenção das lâminas de silício, um resfriamento rápido pode introduzir defeitos no interior do material.

Assim, a temperatura do sólido aquecido precisa ser reduzida a uma velocidade lenta e controlada para assegurar que a estrutura do mesmo tenda para um estado cristalino de energia potencial mínima. A solidificação com um estado cristalino altamente ordenado corresponda ao estado de energia mais baixo (energia interna). Este processo de arrefecimento a uma velocidade lenta é conhecido como recozimento. Quando simulado em computador, recebe o nome de recozimento simulado.

Função objetivo

Quando o recozimento simulado é aplicado para minimização de uma função f , o dual da energia no mundo físico é o valor de uma função $f(X_i)$ a ser minimizada, em que X_i corresponde a um estado do sistema, conforme mostrado na equação (3).

$$E_i = f_i = f(X_i) \quad (3)$$

Critério de Metrópolis

No recozimento simulado, gera-se, a partir de um estado X_i , um estado X_{i+1} . O sistema não passa automaticamente do estado X_i para o estado X_{i+1} . A passagem é controlada através do critério de Metrópolis. O critério de Metrópolis calcula uma probabilidade de

aceitação do novo estado X_{i+1} utilizando a distribuição de Boltzmann. A probabilidade de aceitar o novo estado é dada pela seguinte equação (4):

$$P[E_{i+1}] = \min\{1, e^{-\Delta E/kT}\} \quad (4)$$

Em que:

$$\Delta E = E_{i+1} - E_i = \Delta f = f_{i+1} - f_i \equiv f(X_{i+1}) - f(X_i) \quad (5)$$

Se $\Delta E \leq 0$, então $P[E_{i+1}] = 1$, e, portanto, o novo estado X_{i+1} é aceito incondicionalmente. Isto é uma escolha lógica no contexto da minimização, afinal o valor de $f(X_{i+1})$ é melhor (menor) do que $f(X_i)$. Por outro lado, quando $\Delta E > 0$, $P[E_{i+1}] < 1$, ou seja, a probabilidade de aceitar o novo estado X_{i+1} é menor do que 1. Isso faz sentido, pois o valor de $f(X_{i+1})$ é pior (maior) do que o valor $f(X_i)$. Computacionalmente, a aceitação ou não de X_{i+1} é determinada pela geração de um número aleatório entre 0 e 1. Se o mesmo for menor do que a probabilidade $P[E_{i+1}]$ (também um número entre 0 e 1), aceita-se o novo estado X_{i+1} . Caso contrário, não aceita-se e o algoritmo gera um novo estado X_{i+1} . A redução gradativa da temperatura T no recozimento simulado, torna a aceitação de novos estados cada vez mais difícil, pois as probabilidades de aceitação, $P[E_{i+1}]$ para valores de $\Delta E > 0$, ficam cada vez menores.

Parâmetros

As escolhas da temperatura inicial T_i , o número de iterações n em cada temperatura e o fator de redução da temperatura c desempenham papéis importantes na convergência do algoritmo SA. Por exemplo, se a temperatura inicial T_i for muito grande, o processo de convergência será mais lento. Por outro lado, se a temperatura inicial for muito pequena, o

processo de pesquisa pode deixar de investigar completamente o espaço de soluções, havendo a possibilidade de não pesquisar a região onde se encontra o mínimo global da função. O fator de redução da temperatura c tem um efeito semelhante. Um valor de c entre 0,8 ou 0,9 requer muito esforço computacional para a convergência. Por outro lado, pequenos valores de c (0,1 ou 0,2) podem resultar em uma redução mais rápida na temperatura, não permitindo uma exploração completa do espaço de soluções. O número de iterações n vai ajudar na obtenção de um estado de equilíbrio. Valores maiores de n podem resultar em um esforço computacional maior. Um valor menor de n , por outro lado, pode resultar numa convergência prematura ou convergência para um mínimo local (devido à exploração inadequada do espaço de estados para o mínimo global). Infelizmente, nenhum conjunto único de valores ótimos pode ser pré-fixado para T_i , n ou c .

Algumas orientações podem ser dadas para selecionar esses valores. A temperatura inicial T_i pode ser escolhida como o valor médio da função objetivo calculada em uma série de estados selecionados aleatoriamente no espaço de estados. O número de iterações n pode ser escolhido entre 50 e 100 com base nos recursos de computação e na precisão desejada da solução. O fator de redução da temperatura c pode ser escolhido entre 0,4 e 0,6 para uma estratégia razoável de redução de temperatura.

Qualquer função monotônica decrescente pode ser usada como função de resfriamento. Uma escolha muito usual é a redução linear da temperatura, fazendo $T^{k+1} = \alpha T^k$. Funções exponenciais de resfriamento são também utilizadas.

Estruturas de vizinhança

A geração de uma nova solução, ou um novo estado, é realizada através da geração de vizinhos. Não necessariamente esses vizinhos situam-se em uma região próxima no espaço de estados. Se isso ocorresse, o algoritmo SA produziria apenas uma busca local, o que não é o

caso. Dependendo do problema, diversas estruturas de vizinhança podem ser utilizadas. Uma estrutura de vizinhança é uma operação utilizada para modificar uma solução anterior baseada na vizinhança da mesma. Para um mesmo problema, pode-se utilizar várias estruturas de vizinhança. No âmbito de *VRP*, o trabalho de (Kramer *et al.*, 2016) utiliza algumas dessas estruturas, classificadas em: entre rotas e intrarotas. A seguir temos uma descrição dessas estruturas. Primeiramente as entre rotas:

1. Shift(1,0) – Consiste na transferência de um cliente de uma rota para outra rota.
2. Shift(2,0) – Consiste na transferência de dois clientes, k_1 e k_2 , de uma rota para outra. Neste caso é verificada tanto a transferência do arco (k_1, k_2) quanto a do arco (k_2, k_1) .
3. Swap(1,1) – Consiste na troca de um cliente de uma rota com um cliente de outra rota.
4. Swap(2,1) – Consiste na troca de dois clientes adjacentes, k_1 e k_2 , de uma rota com um cliente, k_3 , de outra rota. Neste caso, também verifica-se da transferência do arco (k_1, k_2) e do arco (k_2, k_1) .
5. Swap(2,2) – Consiste na troca de dois clientes adjacentes, k_1 e k_2 de uma rota com dois clientes adjacentes, k_3 e k_4 , de outra rota. Nesta situação, devem ser consideradas todas as 4 combinações possíveis de trocas entre os arcos formados pelos dois pares de clientes.
6. Cross – Consiste na troca de um segmento de uma rota por um segmento de outra rota. Um segmento de rota é determinado pelo conjunto de clientes localizados no trajeto entre dois clientes.

As estruturas intrarotas apresetadas no trabalho foram:

1. Reinserção – Um cliente é removido da rota e inserido em outra posição.
2. Or-Opt2 – Dois clientes adjacentes são removidos da rota e inseridos em outra posição.

3. Or-Opt3 – Três clientes adjacentes são removidos da rota e inseridos em outra posição.

4. Exchange – Troca entre dois clientes não adjacentes.

Fluxograma básico

O fluxograma básico da técnica de recozimento simulado pode ser visto na Figura 17. As etapas mais sensíveis do método são: a seleção da temperatura inicial (T_i), a definição do esquema de resfriamento, a determinação do número de iterações em cada temperatura (n), e a temperatura final (T_f).

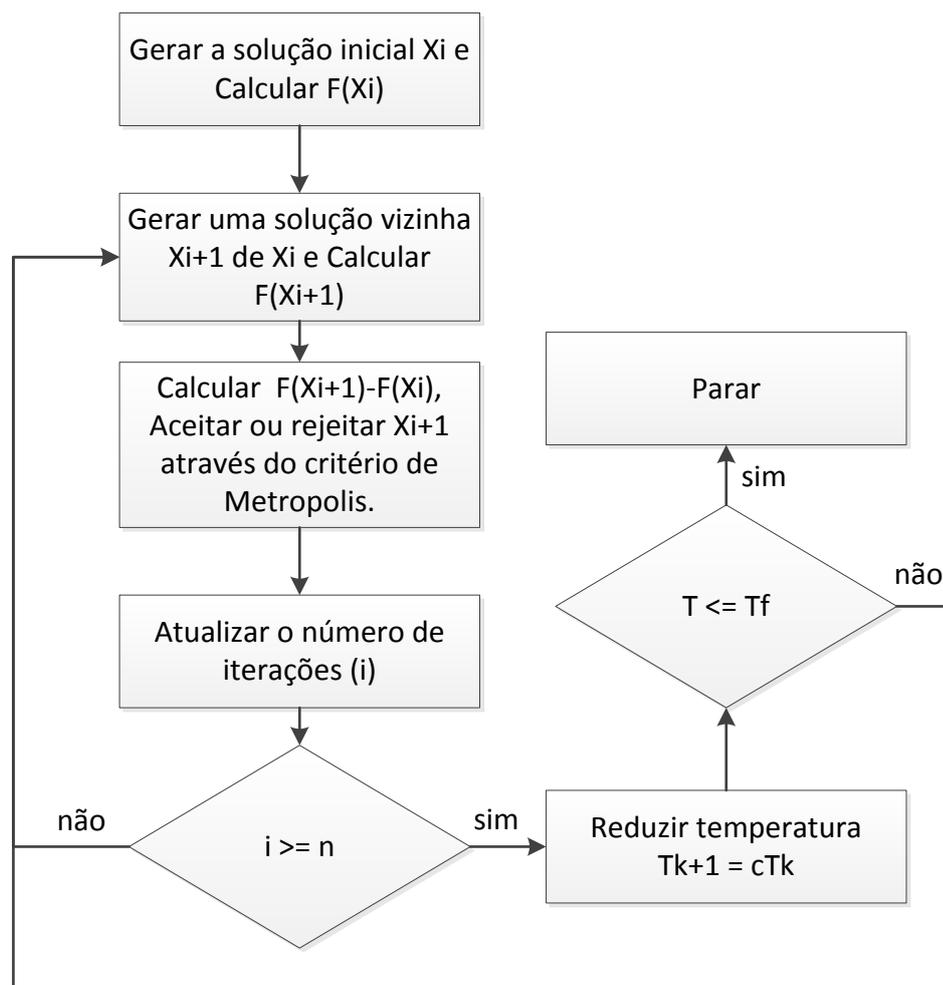


Figura 17 – Fluxograma básico do algoritmo de recozimento simulado.

No trabalho ora proposto, a função objetivo é calculada através de diversas buscas locais com o algoritmo A*, realizada dentro de pequenos agrupamentos. Foi desenvolvida uma nova estrutura de vizinhança para gerar vizinhos, a geração de vizinhos por quadrante. Melhores explicações sobre o desenvolvimento e aplicação dos métodos serão dadas no capítulo de materiais e métodos.

CAPÍTULO 4

MATERIAIS E MÉTODOS

No capítulo anterior apresentou-se os fundamentos teóricos que serão utilizados no desenvolvimento deste trabalho. O capítulo atual descreve a metodologia empregada nesse trabalho com o objetivo de atender ao objetivo principal e aos objetivos específicos delineados na introdução. Nesse sentido, implementou-se um método principal, com todas as contribuições propostas no trabalho e quatro métodos secundários, contendo cada uma das contribuições. A comparação do desempenho do método principal com o desempenho dos métodos secundários permite avaliar o efeito de cada uma das contribuições propostas e ressaltadas nos objetivos específicos. Reafirma-se as principais contribuições desse trabalho: o método de otimização global-local-g; o novo método para geração dos vizinhos para o algoritmo de recozimento simulado e a heurística do passo seguinte para o algoritmo de busca local A*. Esse capítulo detalhará cada uma dessas contribuições e os métodos propostos para a avaliação das mesmas.

4.1 MATERIAIS

O material utilizado nesse trabalho para o auxílio na resolução é o disponibilizado pela Secretaria Municipal de Educação do Município de Manaus (SEMED – Manaus). Essa base de dados é constituída de informações referentes a dados logísticos das escolas pertencentes à Região IV, descrita anteriormente, e de georreferenciamento da cidade de Manaus.

Os dados logísticos consistem em:

1. Veículos de entrega
2. Horário de entrega da merenda escolar

3. Censo escolar do ano de 2016
4. Cardápios por modalidade
5. Per capita da refeição

O horário de entrega da merenda escolar é de 8:00h às 17:30h). O censo escolar é constituído das seguintes informações: nome da escola, modalidade, turnos e capacidade de alunos. Os cardápios e per capita de 54 escolas foram coletados diretamente no setor logístico da SEMED. O veículo de entrega, de gêneros alimentícios não perecíveis, destacado para a região em questão é um caminhão do tipo baú (Tara: 6.590 kg | Líquido: 8.400,00 kg | PTB: 14.990 kg), que possui uma velocidade máxima permitida em área urbana de 60 km/h.

Os dados georreferenciados dizem respeito as localizações dos seguintes itens:

1. Região IV
2. Centro de Distribuição
3. Escolas
4. Logradouros

A Figura 18 mostra os dados georreferenciados de uma forma gráfica para melhor entendimento.



Figura 18: Gráfico mostrando os dados georreferenciados; região IV, escolas, centro de distribuição e escola.

Os dados georreferenciados foram coletados diretamente no setor de georreferenciamento da SEMED. Todas as coordenadas estão em sistema de projeção RTM (*Regional Transverso de Mercator*). A Região IV foi vetorizada de forma que se podem coletar todos os 4.100 pontos de encontro entre os logradouros. Esses pontos coletados são chamados de nodos. As localizações do centro de distribuição e das escolas foram inseridas no mapa como novos nodos. Quanto aos 6.144 logradouros obteve-se nome, localização dos nodos de início e fim, e, o tamanho de cada um.

4.2 MÉTODOS

Conforme ressaltado na introdução, esse trabalho propõe um novo método de otimização para o *VRP*. No capítulo de introdução denominamos esse método de método de otimização global-local-g. Conforme já mencionado, na otimização global utilizou-se o método de recozimento simulado, exposto no item 3.5.1. Na otimização local, utilizou-se o método de busca A^* , exposto no item 3.4.3 Para melhor entendermos as contribuições do trabalho, identificaremos as mesmas fazendo referências aos passos do fluxograma do algoritmo *SA* mostrado na Figura 17. Abaixo listamos essas contribuições:

- 1) Uma forma de determinar o vizinho X_{i+1} de um estado X_i , no (Passo 2) do algoritmo *SA*. Esse novo método é denominado de “geração de vizinhos por quadrante”;
- 2) Uma forma de calcular a função objetivo em um estado X_i (Passo 1), ou em um estado X_{i+1} (Passo 2), denominada de “função objetivo 1”, com duas características principais:
 - a) Utilização das restrições de capacidade ou de tempo, a que ocorrer primeiro, para determinação do tamanho de uma rota;
 - b) Utilização da busca pelo menor caminho A^* para otimizar o percurso de cada rota definida no (item a).
- 3) Proposta de uma heurística para melhoria do desempenho em tempo do algoritmo de busca A^* , denominada de “heurística do passo seguinte”.

A fim de aquilatar o impacto das contribuições 1, 2, e 3, propõe-se cinco versões do algoritmo *SA*. A primeira é denominada de método proposto, e as outras de métodos. Essas versões são descritas a seguir.

Método Proposto: Implementa-se o algoritmo SA utilizando as contribuições 1,2 e 3;

Método GL: Implementa-se o algoritmo SA sem utilizar nenhuma das contribuições.

Método GLG: Implementa-se o algoritmo SA utilizando somente as contribuições 2;

Método QNG: Implementa-se o algoritmo SA utilizando apenas a contribuição 1;

Método QNG_GLG: implementa-se o algoritmo SA utilizando as contribuições 1 e 2.

A Figura 19 mostra o método proposto e os quatro métodos ilustrados para melhor compreensão.

Para fins de comparação, nos métodos QNG e QNG_GLG, anteriormente citados, o desempenho do algoritmo A* é comparado com o desempenho do algoritmo Dijkstra.

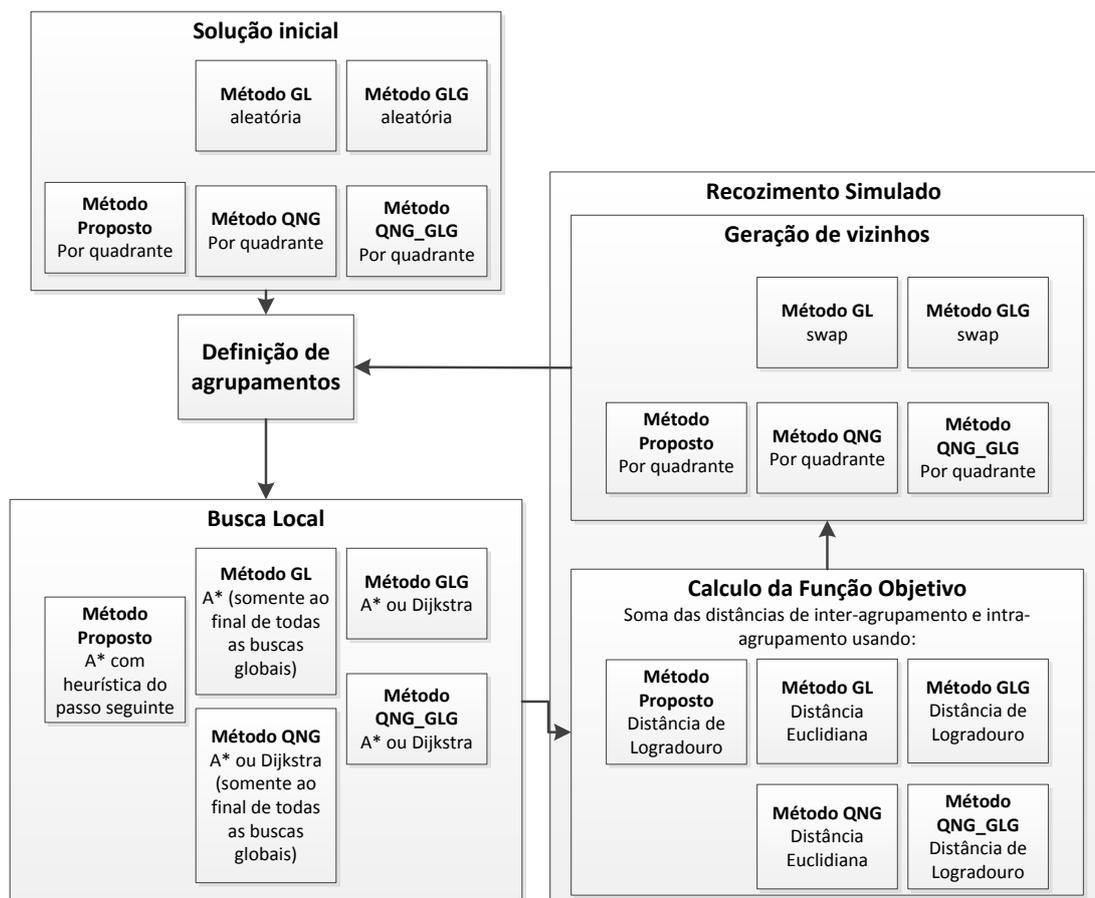


Figura 19 – Fluxograma mostrando as principais características do Método Proposto e dos métodos. O termo por quadrante refere-se ao método de geração de vizinhos por quadrante.

Os métodos GL e QNG, utilizam apenas distâncias euclidianas (em linha reta) para o cálculo da função objetivo, chamada de função objetivo 2. As rotas são determinadas somente pela restrição de capacidade, ou seja, só se conhece a distância real de uma escola para outra após a busca local, essa distância real foi denominada de “distância de logradouro”. Apenas após as buscas globais, utiliza-se a busca A* para uma otimização final. No capítulo de introdução e ao final do capítulo 2 denominamos esse método de busca de método de busca global-local.

No método proposto e nos métodos GLG e QNG_GLG, para cada candidato gerado na busca global são realizadas “g” buscas locais. A letra g refere-se ao número de agrupamentos de escolas geradas pelas restrições de capacidade ou tempo. No capítulo de introdução e no final do capítulo 2 denominamos esse método de busca de global-local-g.

Na sequência, descreveremos de maneira mais aprofundada a modelagem do problema a ser resolvido, o modelo de codificação da solução, o método de geração de vizinhos por quadrante, o cálculo da função objetivo 1 e a heurística do passo seguinte.

4.2.1 Problema de roteamento de veículos com janela de tempo

O problema de roteamento de veículos com janela de tempo (VRPTW) é definido por um grafo $G = (E, A)$, onde: $E = \{0, \dots, n\}$ corresponde a um conjunto de vértices em que, tradicionalmente, o vértice 0 é chamado de depósito, enquanto os vértices restantes em E representam as escolas (ou pedidos) que precisam ser atendidos. Todas as rotas iniciam e terminam no depósito e_0 ; $A = \{(i, j) | i \in E, j \in E, i \neq j\}$ corresponde a um conjunto de arcos; Neste contexto, o custo c_{ij} (distância entre dois vértices i e j) e um tempo de viagem t_{ij} são associados a cada arco $(i, j) \in A$. O tempo de viagem da escola i até a escola j é representado por t_{ij} e inclui o tempo de serviço na escola i . Um conjunto de veículos idênticos é dominado V , onde cada veículo possui capacidade Q , e a cada escola $i \in E$ é atribuída uma demanda d_i ,

a demanda d_0 do depósito é nula. Em cada escola, o início do serviço precisa ser realizado em um certo intervalo de tempo chamado de janela de tempo, $[a_i, b_i]$, $i \in E$. Os veículos devem retornar ao depósito durante a janela de tempo $[a_0, b_0]$. Um veículo é permitido chegar antes da abertura da janela de tempo, e esperar até que o serviço possa ser efetuado, mas não é permitido chegar após a janela de tempo. Desde que a espera seja sem custo, podemos assumir que $a_0 = b_0 = 0$, ou seja, todas as rotas iniciam-se no tempo 0.

O modelo contém duas variáveis de decisão. A primeira, X_{ij}^k , definida para $\forall(i, j) \in E, \forall k \in V$, é igual a 1 se o veículo k percorre o caminho da escola i até a escola j , e 0 caso contrário. A segunda, S_i^k , definida por $\forall i \in E, \forall k \in V$, representa o tempo em que o veículo k chega na escola i . Se um certo veículo k não realiza serviço na escola i , S_i^k é igual a 0. A soma de todas as demandas d_i de todas as escolas servidas por um veículo deve ser menor do que a capacidade do veículo Q . O objetivo é minimizar o percurso da rota de cada veículo, tal que todas as escolas sejam servidas exatamente uma única vez; logo, não são permitidas entregas separadas. As rotas devem ser viáveis em relação à capacidade dos veículos e às janelas de tempo dos clientes atendidos. Abaixo descrevemos esse problema matematicamente:

Minimizar

$$\sum_{k \in V} \sum_{(i,j) \in A} c_{ij} X_{ij}^k \quad (1)$$

Sujeito a

$$\sum_{k \in V} \sum_{j \in E} X_{ij}^k = 1, \forall i \in E \quad (2)$$

$$\sum_{i \in E} \sum_{j \in E} (d_i + d_j) X_{ij}^k \leq Q, \forall k \in V \quad (3)$$

$$\sum_{j \in E} X_{0j}^k = 1, \forall k \in V \quad (4)$$

$$\sum_{i \in E} X_{ih}^k - \sum_{j \in E} X_{hj}^k = 0, \forall h \in E, \forall k \in V \quad (5)$$

$$\sum_{i \in E} X_{i,0}^k = 1, \forall k \in V \quad (6)$$

$$X_{ij}^k (S_i^k - t_{ij} - S_j^k) \leq 0, \forall (i,j) \in A, \forall k \in V \quad (7)$$

$$a_0 + \sum_{i \in E} \sum_{j \in E} t_{ij} X_{ij}^k, \forall k \in V \leq b_0 \quad (8)$$

$$X_{ij}^k \in \{0,1\}, \forall (i,j) \in A, \forall k \in V \quad (9)$$

Primeiramente, a função objetivo (1) denota que os custos que devem ser minimizados. O conjunto de restrições (2) determina que cada escola deve ser atribuída exatamente a um veículo. O conjunto de restrições (3) determina que nenhum veículo pode atender mais escolas do que sua capacidade permite. O conjunto de restrições (4), (5), e (6) são as restrições de fluxo, que determinam que cada veículo k saia do nodo 0 (depósito) uma vez, deixe o nodo h , onde $h \in E$, se e somente se ele passa por esse nodo, e, retorna ao nodo 0. Observe que o conjunto de restrições (6) é redundante, mas é mantido no modelo para fechar a estrutura da rede. O conjunto de restrições (7) indica que o veículo k viaja de i para j , e, o mesmo não pode chegar a j antes de decorrido o tempo $S_i + t_{ij}$. O conjunto de restrições (8) garante que todas as janelas de tempo são respeitadas. Finalmente, o conjunto de restrições (9) é o conjunto de restrições de integralidade.

4.2.2 Codificação da solução

A codificação da solução é diferente das apresentadas em outros trabalhos, pois não utiliza somente os nodos de interesse, ou seja, as escolas, mas utiliza também os nodos que representam os cruzamentos entre as ruas que são percorridos na ligação de uma escola a outra, isso faz com que o motorista além de ter a sequência de escolas a ser seguida, também tenha o caminho a ser percorrido entre essas escolas. Para um melhor entendimento da codificação do problema apresentamos a Figura 20. Nessa figura D representa o depósito, E representa uma escola, e N representa cruzamentos entre ruas. A Figura 20(a) mostra a

representação de duas rotas que se iniciam no depósito. A rota 1 é D0- E1- E2-D0. A rota 2 é D0-E3-E4-D0. Essa representação, contendo apenas os nodos das escolas, é a utilizada pelo algoritmo de busca global. A Figura 20(b) mostra as mesmas duas rotas da Figura 20(a). Entretanto, a mesma mostra não só as escolas, mas os nodos referentes aos cruzamentos (N_i , $1 < i < 27$) das ruas situadas no caminho entre duas escolas. Essa representação, contendo os nodos das escolas e dos cruzamentos, é a representação utilizada no algoritmo de busca local. Na solução ótima esses pontos descrevem a menor rota a ser percorrida. A rota 1 agora é representada como D0-N6-N7-E1-N1-N2-N3-N4-E2-N19-N20-D0, e a rota 2 como D0-N22-N12-N21-N13-E3-N8-N9-N10-E4-N23-N24-N25-N26-N27-D0.

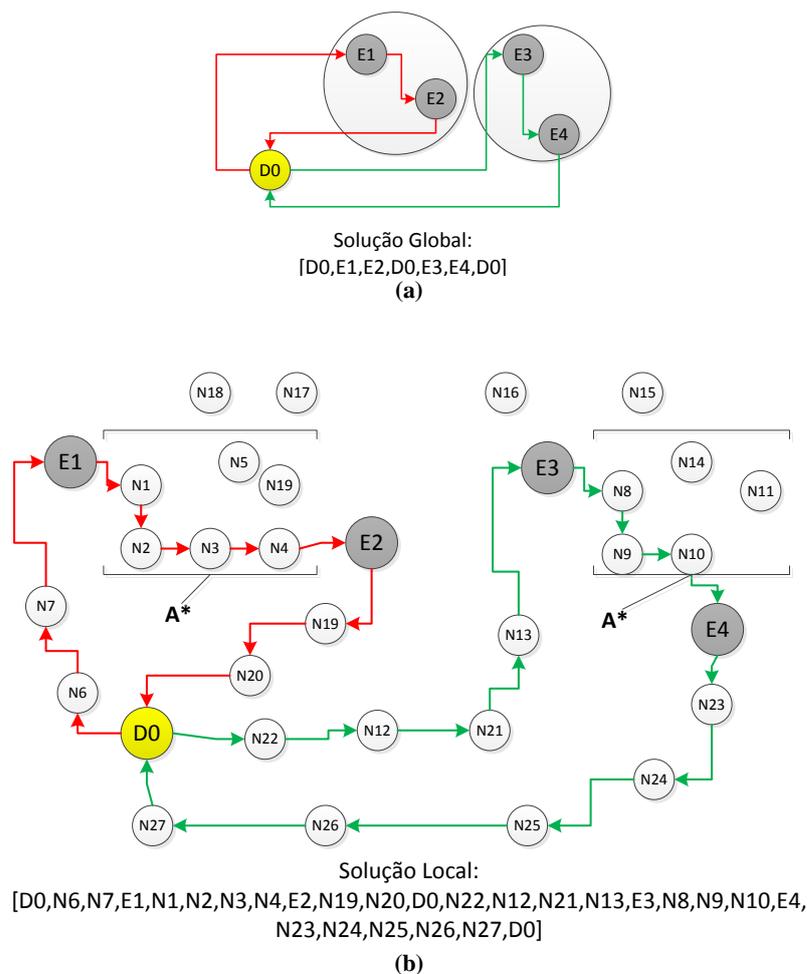


Figura 20 – (a) Representações utilizadas no algoritmo de Busca Global – Recozimento Simulado. (b) Representação utilizada nos algoritmos de Busca Local – Algoritmo de busca A* ou Dijkstra

4.2.3 Geração de vizinhos por quadrante

A aplicação do método dos vizinhos por quadrante baseia-se na divisão da região onde o *VRP* será resolvido. Para isso utilizaremos três fases: divisão em quadrantes, atribuição das escolas aos quadrantes e, por último a geração de vizinhos.

Divisão em quadrantes

Primeiramente é necessário encontrar o raio e ângulo de cada escola localizada nessa região. Para exemplificar utilizaremos a Figura 21, com o círculo amarelo representando o centro de distribuição, localizado nas coordenadas (x_0, y_0) , e o círculo vermelho uma escola i da região IV. As coordenadas são polares, considerando o centro de distribuição como a origem do sistema de coordenadas. A localização da escola é dada por: (r_i, θ_i) ,

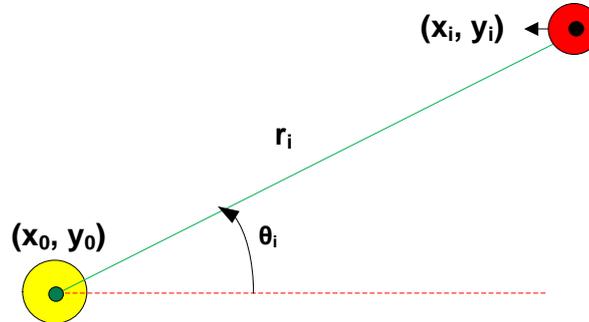


Figura 21 – Coordenadas polares e cartesianas de uma escola

em que o raio:

$$r_i = \sqrt{X_i^2 + Y_i^2} \quad (10)$$

E o ângulo,

$$\theta_i = \begin{cases} \tan^{-1} \frac{y_i}{x_i}, & X_i > 0, Y_i > 0 \\ \pi + \tan^{-1} \frac{y_i}{x_i}, & X_i < 0 \\ 2\pi + \tan^{-1} \frac{y_i}{x_i}, & X_i > 0, Y_i < 0 \end{cases} \quad (11)$$

Sendo que:

$$X_i = x_i - x_o \quad (12)$$

$$Y_i = y_i - y_o \quad (13)$$

Após o cálculo das coordenadas polares de todas as escolas pertencentes à região de interesse, calcula-se os limitadores dos quadrantes. A divisão em quadrantes da região onde o *VRP* é resolvido é mostrada na Figura 22. Os quadrantes são numerados no sentido anti-horário, iniciando a numeração no canto inferior esquerdo. Nessa figura, os retângulos verdes representam as escolas. Os limites dos quadrantes são definidos pelos valores de raio máximo (r_{\max}), raio mínimo (r_{\min}), ângulo máximo (θ_{\max}) e ângulo mínimo (θ_{\min}) das escolas. Os limites internos dos quadrantes são definidos através dos valores do ângulo médio ($\theta_{\text{médio}}$) e do raio médio ($r_{\text{médio}}$). O raio médio é calculado utilizando a seguinte equação:

$$r_{\text{medio}} = \frac{r_{\min} + r_{\max}}{2} \quad (14)$$

O ângulo médio é calculado utilizando a equação:

$$\theta_{\text{medio}} = \frac{\theta_{\min} + \theta_{\max}}{2} \quad (15)$$

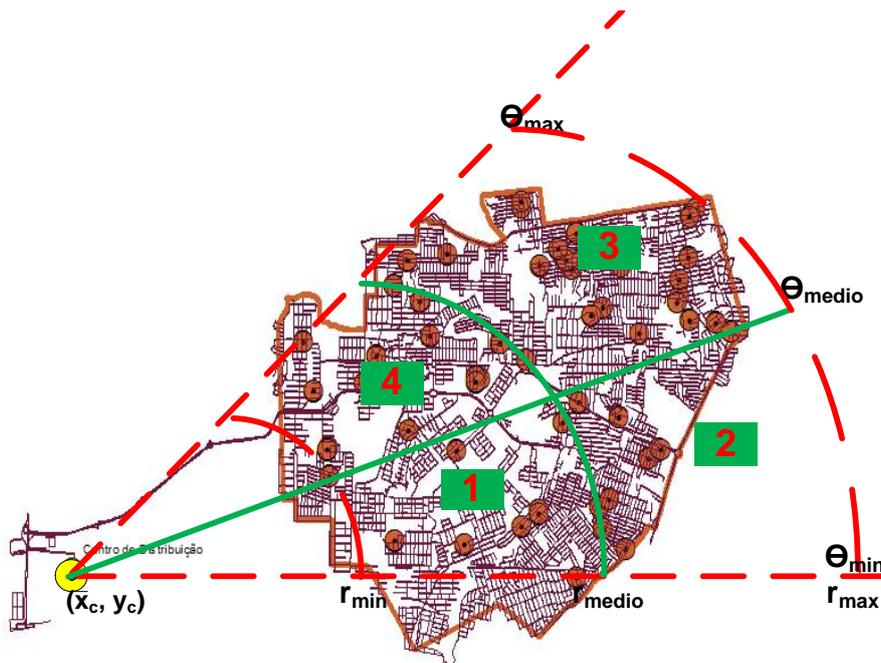


Figura 22 – Região IV dividida em quadrantes.

Atribuição das escolas aos quadrantes

A atribuição das escolas aos quadrantes é feita comparando-se as coordenadas polares das escolas com os limites dos quadrantes. No método de geração de vizinhos por quadrante, esses limites não são fixos. Para cada novo vizinho muda-se os valores dos mesmos. Isso fará com que as escolas mantenham-se em constante mudança de quadrantes. A Rotina 2, que será vista na fase de geração de vizinhos, mostrará como as escolas são atribuídas aos seus quadrantes.

Geração de vizinhos

A descrição do método de geração das soluções vizinhas será feita seguindo o fluxograma mostrado na Figura 23 e com o auxílio das Figuras 24, 25, 26, 27, que descrevem as rotinas, e a Figura 28 que descreve as sequências de giro. Como veremos, um vizinho é gerado definindo um quadrante inicial e uma sequência ou *tour* para se percorrer os quadrantes. Dentro de cada quadrante as escolas são percorridas usando o método do vizinho mais próximo. No início do *tour*, utiliza-se o vizinho mais próximo ao depósito, esse vizinho deve pertencer ao quadrante inicial definido anteriormente. As rotas são definidas seguindo a ordem dos quadrantes e dentro de cada quadrante aplica-se o método do vizinho mais próximo.

Para a geração do primeiro vizinho do método SA, conforme mostrado no Bloco 1 do Fluxograma da Figura 23, calculamos os valores de r_{medio} e θ_{medio} , conforme definido nas equações (14) e (15). Na geração dos outros vizinhos para os valores de r_{medio} e θ_{medio} é utilizado o procedimento descrito na Rotina 1, Figura 24. Nesse procedimento, os valores de r_{medio} e θ_{medio} sofrem incrementos ou decrementos de 10% em cima do seu valor atual de cada um deles. Caso o limite máximo (r_{max}) seja atingido, o quadrante de início para a geração de um vizinho passa a ser o quadrante seguinte àquele definido como sendo o

quadrante de início do *tour* e o valor de r_{medio} é ajustado conforme mostrado na Rotina 1. Caso o limite mínimo de r_{min} seja atingido, o quadrante de início para a geração de um vizinho passa a ser o quadrante anterior àquele definido como sendo o quadrante de início do *tour* e o valor de r_{medio} é ajustado conforme mostrado na Rotina 1. Caso o limite máximo de θ_{max} seja atingido, o valor de θ_{medio} é ajustado conforme mostrado na Rotina 1. Caso o limite mínimo de θ_{min} seja atingido, o valor de θ_{medio} é ajustado conforme mostrado na Rotina 1. Do que foi exposto nesse parágrafo, conclui-se que os valores de r_{medio} e θ_{medio} não são fixos. A partir da Figura 22 observa-se que os valores dessas duas variáveis determinam os limites internos dos quadrantes. Assim, conclui-se que os tamanhos dos quadrantes são variáveis.

No Bloco 2 do Fluxograma da Figura 23, atribui-se cada escola a um quadrante. Isso é feito utilizando-se a Rotina 2, Figura 25.

Entre os Blocos 2 e 3, se o algoritmo *SA* encontra-se no início de um laço interno ($i = 0$), define-se um quadrante para o início do *tour* e a orientação de giro do *tour*. Se não utilizam-se os valores definidos no início do laço interno.

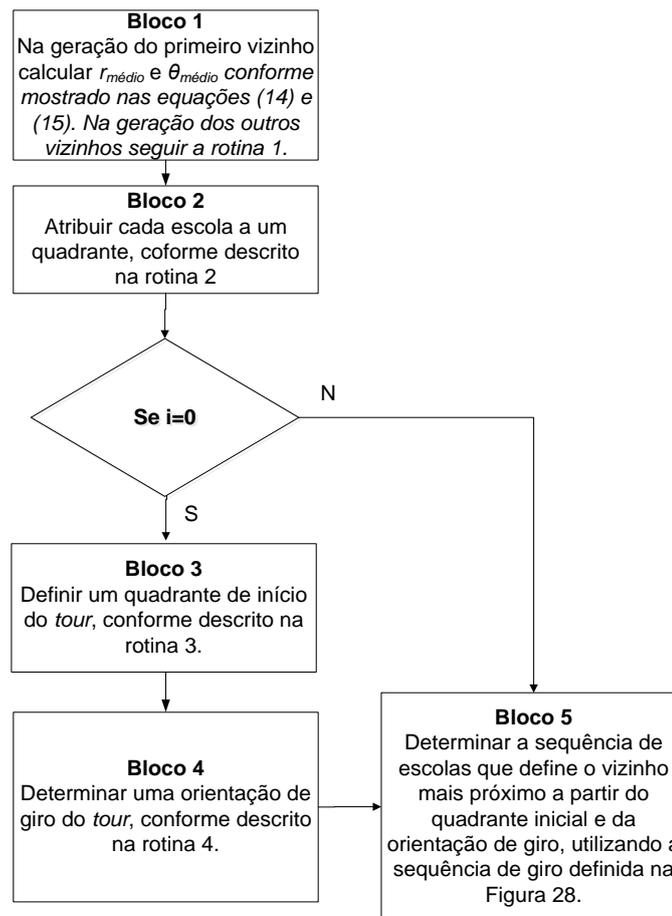


Figura 23 – Fluxograma de geração de vizinhos por quadrantes no método SA

No Bloco 3, define-se o quadrante de início do *tour*, ou seja, define-se um quadrante inicial para percorrer o *tour* gerando um novo estado vizinho. Isso é efetuado utilizando-se a Rotina 3 da Figura 26. Conforme mostrado nessa rotina, cada um dos quatro quadrantes pode ser o quadrante de início do *tour*.

No Bloco 4 define-se a orientação de giro do *tour*. Isso é feito utilizando-se a Rotina 4 da Figura 27. Como mostrado nessa rotina, a orientação de giro pode assumir um entre seis valores: horário, anti-horário, diagonal horário, diagonal anti-horário, horário diagonal anti-horário, anti-horário diagonal horário. Na Figura 28 mostram-se os quadrantes percorridos para cada um destes sentidos em função do quadrante definido como sendo o de início do *tour*. Vamos supor, por exemplo, que no Bloco 3 determinou-se que o quadrante de início do *tour* seria o quadrante 2 e que no Bloco 4 determinou-se que a orientação de giro é a anti-

horário. Então, conforme mostrado na Figura 28, na determinação do vizinho, os quadrantes serão percorridos na sequência 2, 3, 4 e 1.

Rotina 1 – Cálculo de um novo $r_{médio}$ e $\theta_{médio}$
 Gerar um número aleatório m entre 0 e 1
 Se $m > 0,5$
 então fazer $r_{médio} = 1,1 * r_{médio}$
 Se $r_{médio} \geq r_{max}$
 Quadrante de início = quadrante seguinte na sequência de giro
 $r_{médio} = 0,75r_{max}$
 Caso contrário
 então fazer $r_{médio} = 0,9 * r_{médio}$
 Se $r_{médio} \leq r_{min}$
 Quadrante de início = quadrante anterior na sequência de giro
 $r_{médio} = 1,25r_{min}$
 Gerar um número aleatório n entre 0 e 1
 Se $n > 0,5$
 então fazer $\theta_{médio} = 1,1 * \theta_{médio}$
 Se $\theta_{médio} \geq \theta_{max}$
 Quadrante de início = quadrante anterior na sequência de giro
 $\theta_{médio} = 1,25r_{min}$
 Caso contrário
 então fazer $\theta_{médio} = 0,9 * \theta_{médio}$
 Se $\theta_{médio} \leq \theta_{min}$
 Quadrante de início = quadrante anterior na sequência de giro
 $\theta_{médio} = 0,75\theta_{min}$

Figura 24 – Rotina para manter os parâmetros ($r_{médio}$, $\theta_{médio}$) no interior da área definida

Rotina 2 – Determinação do quadrante de cada escola i :
 Gerar $r_{médio}$ e $\theta_{médio}$
 Se $r_{min} \leq r_i < r_{médio}$ e $\theta_{min} \leq \theta_i < \theta_{médio}$
 Atribuir a escola i ao quadrante 1;
 Se $r_{médio} \leq r_i < r_{max}$ e $\theta_{min} \leq \theta_i < \theta_{médio}$
 Atribuir a escola i ao quadrante 2;
 Se $r_{médio} \leq r_i < r_{max}$ e $\theta_{médio} \leq \theta_i < \theta_{max}$
 Atribuir a escola i ao quadrante 3;
 Se $r_{min} \leq r_i < r_{médio}$ e $\theta_{médio} \leq \theta_i < \theta_{max}$
 Atribuir a escola i ao quadrante 4;

Figura 25 – Rotina para atribuir as escolas aos quadrantes

Rotina 3 – Determinação do quadrante de início para a definição do vizinho:
 Gerar aleatoriamente r entre r_{min} e r_{max} e θ entre θ_{min} e θ_{max}
 Se $r_{min} \leq r < r_{médio}$ e $\theta_{min} \leq \theta < \theta_{médio}$
 Iniciar a *tour* pelo quadrante 1
 Se $r_{médio} \leq r < r_{max}$ e $\theta_{min} \leq \theta < \theta_{médio}$
 Iniciar a *tour* pelo quadrante 2
 Se $r_{médio} \leq r < r_{max}$ e $\theta_{médio} \leq \theta < \theta_{max}$
 Iniciar o *tour* pelo quadrante 3
 Se $r_{min} \leq r < r_{médio}$ e $\theta_{médio} \leq \theta < \theta_{max}$
 Iniciar o *tour* pelo quadrante 4

Figura 26 – Rotina para determinar o quadrante de início do *tour*

Rotina 4 – Determinação do giro do *tour*

Gerar um número n aleatório entre 0 e 1.

$$\text{Se } 0 \leq n < \frac{1}{n.\text{orientações}}$$

Giro é horário.

$$\text{Se } \frac{1}{n.\text{orientações}} \leq n < \frac{2}{n.\text{orientações}}$$

Giro é anti-horário.

$$\text{Se } \frac{2}{n.\text{orientações}} \leq n < \frac{3}{n.\text{orientações}}$$

Giro é diagonal horário.

$$\text{Se } \frac{3}{n.\text{orientações}} \leq n < \frac{4}{n.\text{orientações}}$$

Giro é diagonal anti-horário.

$$\text{Se } \frac{4}{n.\text{orientações}} \leq n < \frac{5}{n.\text{orientações}}$$

Giro é horário diagonal anti-horário.

$$\text{Se } \frac{5}{n.\text{orientações}} \leq n \leq 1$$

Giro é anti-horário diagonal horário.

Em que:

$$n.\text{orientações} = 6.$$

Figura 27 – Rotina para determinar a orientação de giro de *tour*

Horário	Anti-horário	Diagonal Horário	Diagonal Anti-horário	Horário Diagonal Anti-horário	Anti-horário Diagonal horário
$1 \rightarrow 4$ $4 \rightarrow 3$ $3 \rightarrow 2$	$1 \rightarrow 2$ $2 \rightarrow 3$ $3 \rightarrow 4$	$1 \rightarrow 3$ $3 \rightarrow 2$ $2 \rightarrow 4$	$1 \rightarrow 3$ $3 \rightarrow 4$ $4 \rightarrow 2$	$1 \rightarrow 4$ $4 \rightarrow 2$ $2 \rightarrow 3$	$1 \rightarrow 2$ $2 \rightarrow 4$ $1 \rightarrow 3$

(a)

Horário	Anti-horário	Diagonal Horário	Diagonal Anti-horário	Horário Diagonal Anti-horário	Anti-horário Diagonal horário
$2 \rightarrow 1$ $1 \rightarrow 4$ $4 \rightarrow 3$	$2 \rightarrow 3$ $3 \rightarrow 4$ $4 \rightarrow 1$	$2 \rightarrow 4$ $4 \rightarrow 3$ $3 \rightarrow 1$	$2 \rightarrow 4$ $4 \rightarrow 1$ $1 \rightarrow 3$	$2 \rightarrow 1$ $1 \rightarrow 3$ $3 \rightarrow 4$	$2 \rightarrow 3$ $3 \rightarrow 1$ $1 \rightarrow 4$

(b)

Horário	Anti-horário	Diagonal Horário	Diagonal Anti-horário	Horário Diagonal Anti-horário	Anti-horário Diagonal horário
$3 \rightarrow 2$ $2 \rightarrow 1$ $1 \rightarrow 4$	$3 \rightarrow 4$ $4 \rightarrow 1$ $1 \rightarrow 2$	$3 \rightarrow 1$ $1 \rightarrow 4$ $4 \rightarrow 2$	$3 \rightarrow 1$ $1 \rightarrow 2$ $2 \rightarrow 4$	$3 \rightarrow 2$ $2 \rightarrow 4$ $4 \rightarrow 1$	$3 \rightarrow 4$ $4 \rightarrow 2$ $2 \rightarrow 1$

(c)

Horário	Anti-horário	Diagonal Horário	Diagonal Anti-horário	Horário Diagonal Anti-horário	Anti-horário Diagonal horário
$4 \rightarrow 3$ $3 \rightarrow 2$ $2 \rightarrow 1$	$4 \rightarrow 1$ $1 \rightarrow 2$ $2 \rightarrow 3$	$4 \rightarrow 2$ $2 \rightarrow 1$ $1 \rightarrow 3$	$4 \rightarrow 2$ $2 \rightarrow 3$ $3 \rightarrow 1$	$4 \rightarrow 3$ $3 \rightarrow 1$ $1 \rightarrow 2$	$4 \rightarrow 1$ $1 \rightarrow 3$ $3 \rightarrow 2$

(d)

Figura 28 – Sequência em que os quadrantes são percorridos em função do giro e do quadrante de início: (a) origem no quadrante 1, (b) origem no quadrante 2, (c) origem no quadrante 3, (d) origem no quadrante 4

A seguir ilustramos a técnica de geração de vizinhos por quadrante através de um exemplo. Essa técnica é utilizada no método proposto e nos métodos 3 e 4, como pode ser visto na Figura 19. Para essa técnica utilizaremos um total de 15 escolas (a localização dessas escolas é mostrada na Figura 29) com demandas diferentes, entre $141,84 \text{ kg} \leq d \leq 573,49 \text{ kg}$, e 1 veículo com capacidade de 2000 kg. Para a resolução desse *VRP* seguimos os seguintes passos:

- 1) Através das equações (10) e (11) calculamos as coordenadas polares das escolas (r_i, θ_i) , com $1 \leq i \leq 15$;
- 2) Através das equações (14) e (15) calculamos os valores iniciais de r_{medio} e θ_{medio}
 $r_{medio} = 8247,5926217 \text{ m}$, $\theta_{medio} = 0,311920846888 \text{ rad}$;
- 3) Através das coordenadas polares das escolas, definem-se os limites dos quadrantes.

$$r_{min} = 6.181,43202008 \text{ m}, \theta_{min} = 0,0861713192413 \text{ rad},$$

$$r_{max} = 10.138,9057838 \text{ m}, \theta_{max} = 0,526787366847 \text{ rad};$$

- 4) Para simplificarmos o exemplo analisaremos a geração dos vizinhos utilizando 2 valores de temperatura ($T=1$ e $T=0,5$) no algoritmo de recozimento simulado. Em cada temperatura serão utilizadas 3 gerações:

4.1) **T = 1** (laço externo): Utilizando a **Rotina 4**: $n = 0,85$. Logo, orientação de giro do tour = [6], ‘Anti-horário Diagonal Horário’;

4.1.1) Através da **Rotina 3** determinar o quadrante de início do *tour*:

$$r = 7099 \text{ e } \theta = 0,1014.$$

Quadrante de início = 1;

4.1.2) Sequência dos quadrantes do *tour* = [1,2,4,3];

4.1.3) Escola de início do *tour* = 4, pois é a escola do quadrante de início mais próxima ao depósito 0.

4.1.4) Laço interno (método do vizinho mais próximo):

Geração = 1 (solução inicial/T=1): Figura 29

$$r_{medio} = 8247,5926217 \text{ m}, \theta_{medio} = 0,311920846888 \text{ rad.}$$

Solução inicial= [0 4 5 7 9 8 12 6 13 15 11 10 14 3 1 2].

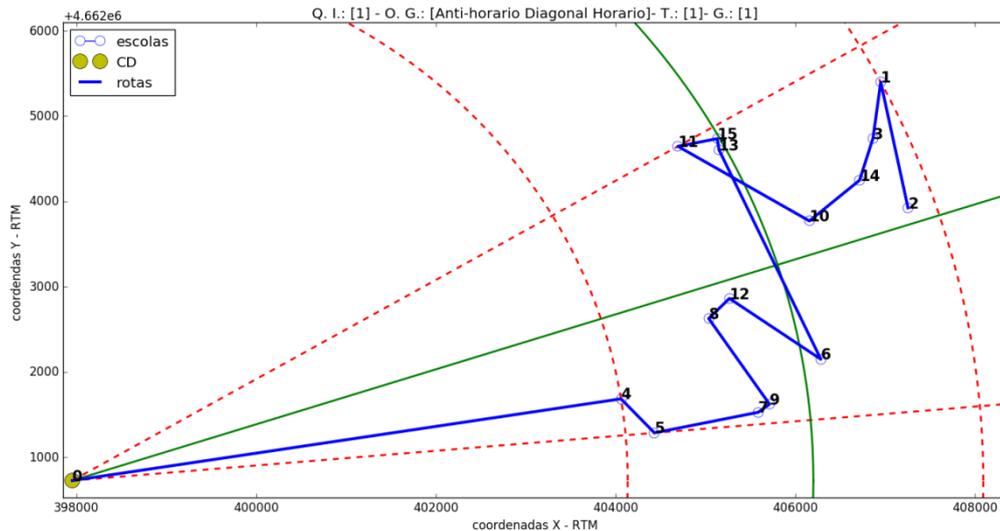


Figura 29 – Solução para T=1, Geração=1, Quadrante de início=1, Orientação de giro = Anti-horário Diagonal Horário, [0 4 5 7 9 8 12 6 13 15 11 10 14 3 1 2].

Utilizando a **Rotina 1** para encontrarmos o próximo raio e ângulo:

$m = 0,1, n = 0,79$: Logo, o raio diminui e o ângulo aumenta.

Geração = 2 (vizinho da geração 1/T=1): Figura 30

$$r_{medio} = 7422,83335953 \text{ m}, \theta_{medio} = 0,343112931577 \text{ rad.}$$

Quadrante de início = 1.

Sequência dos quadrantes do *tour* = [1,2,4,3].

Solução = [0 4 5 8 12 6 9 7 10 14 3 1 2 13 15 11].

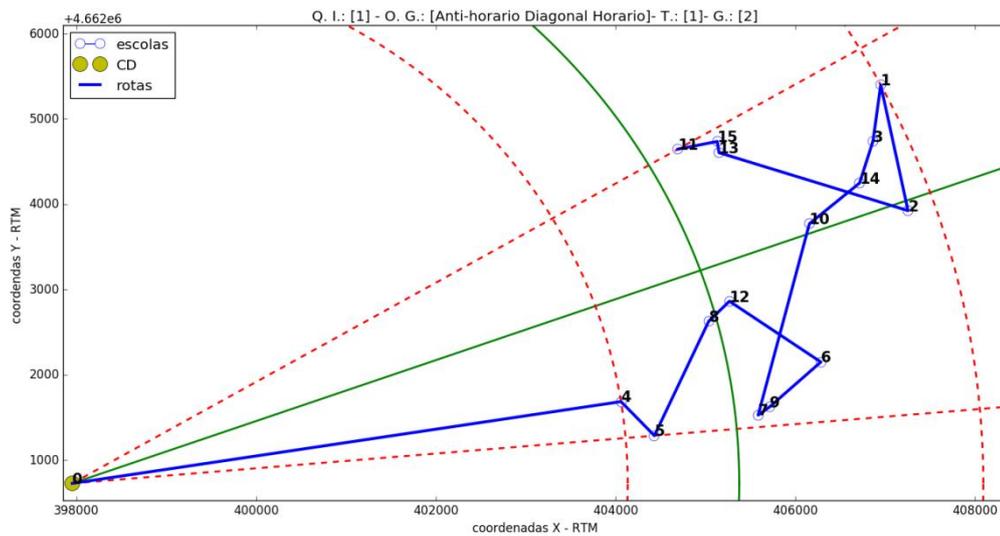


Figura 30 – Solução para $T=1$, Geração=2, Quadrante de início =1, Orientação de giro = Anti-horário Diagonal Horário, [0 4 5 8 12 6 9 7 10 14 3 1 2 13 15 11].

Utilizando a **Rotina 1** para encontrarmos o próximo raio e ângulo:

$m = 0,9, n = 0,17$: Logo, o raio aumenta e o ângulo diminui.

Geração = 3 (vizinho da geração $2/T=1$): Figura 31

$r_{medio} = 8165,11669549 \text{ m}, \theta_{medio} = 0,308801638419 \text{ rad}$

Solução = [0 4 5 7 9 8 12 6 11 15 13 10 14 3 1 2]

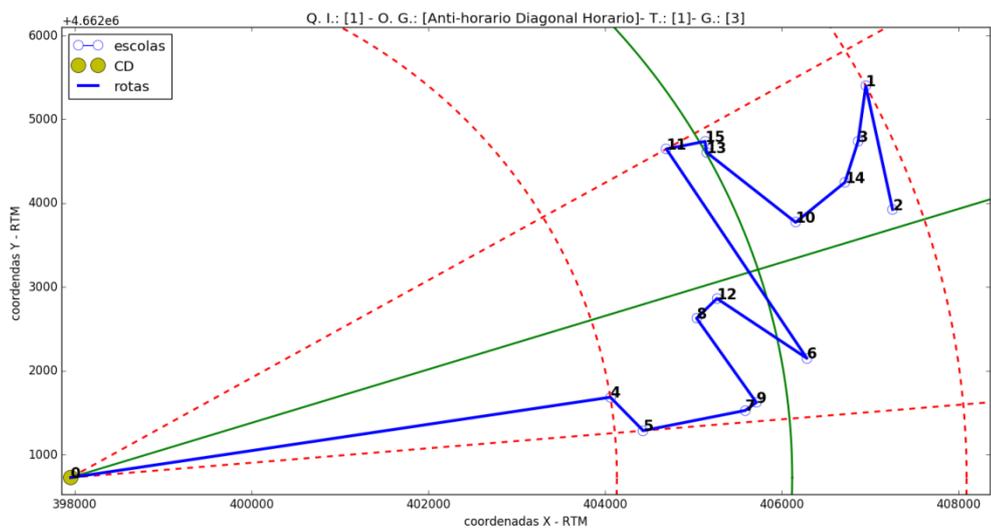


Figura 31 – Solução para $T=1$, Geração=3, Quadrante de início =1, Orientação de giro = Anti-horário Diagonal Horário, [0 4 5 7 9 8 12 6 11 15 13 10 14 3 1 2].

Utilizando a **Rotina 1** para encontrarmos o próximo raio e ângulo:

$m = 0,19, n = 0,98$: Logo, o raio diminui e o ângulo aumenta.

4.2) **T = 0,5** (laço externo): Utilizando a **Rotina 4**: $n = 0,67$, logo, orientação de giro do tour = [5], ‘Horário Diagonal Anti-horário’;

4.2.1) Através da **Rotina 3** determinar o quadrante de início do *tour*:

$$r = 7395 \text{ e } \theta = 0,1449.$$

Quadrante de início = 2;

4.2.2) Sequência dos quadrantes do *tour* = [2,1,3,4];

4.2.3) Escola de início do *tour* = 6, pois é ela a mais próxima do depósito 0.

4.2.4) Laço interno (método do vizinho mais próximo):

Geração = 1 (solução inicial/T=0,5): Figura 32

$$r_{medio} = 7348,60502594 \text{ m}, \theta_{medio} = 0,339681802261 \text{ rad}.$$

Solução = [0 6 9 7 12 8 4 5 10 14 3 1 2 13 15 11].

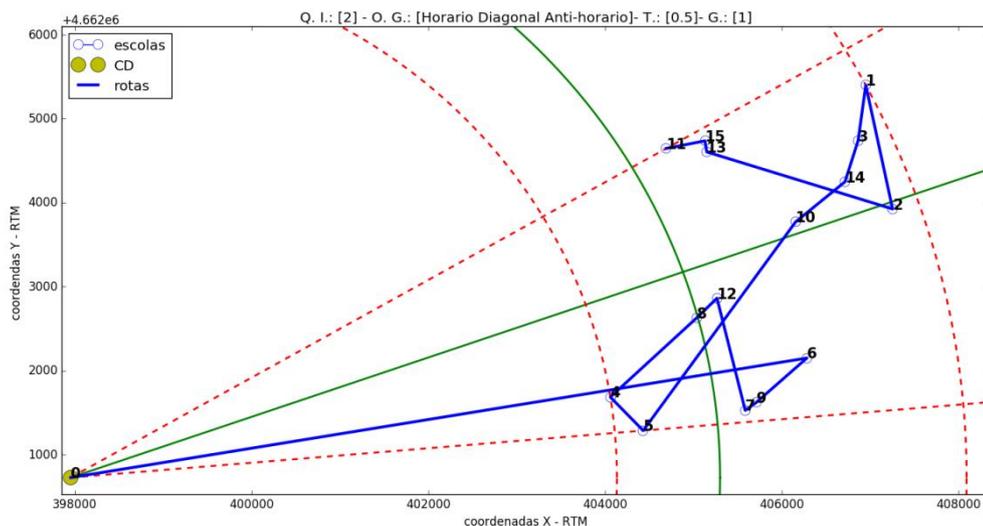


Figura 32 – Solução para T=0,5, Geração=1, Quadrante de início =2, Orientação de giro = Horário Diagonal Anti-horário, [0 6 9 7 12 8 4 5 10 14 3 1 2 13 15 11].

Utilizando a **Rotina 1** para encontrarmos o próximo raio e ângulo:
 $m = 0,25, n = 0,94$: Logo, o raio diminui e o ângulo aumenta.

Geração = 2 (vizinho da geração $1/T=0,5$): Figura 33

$$r_{medio} = 6613,74452334 \text{ m}, \theta_{medio} = 0,373649982487 \text{ rad}$$

Solução = [0 6 9 7 8 12 4 5 10 14 3 1 2 13 15 11]

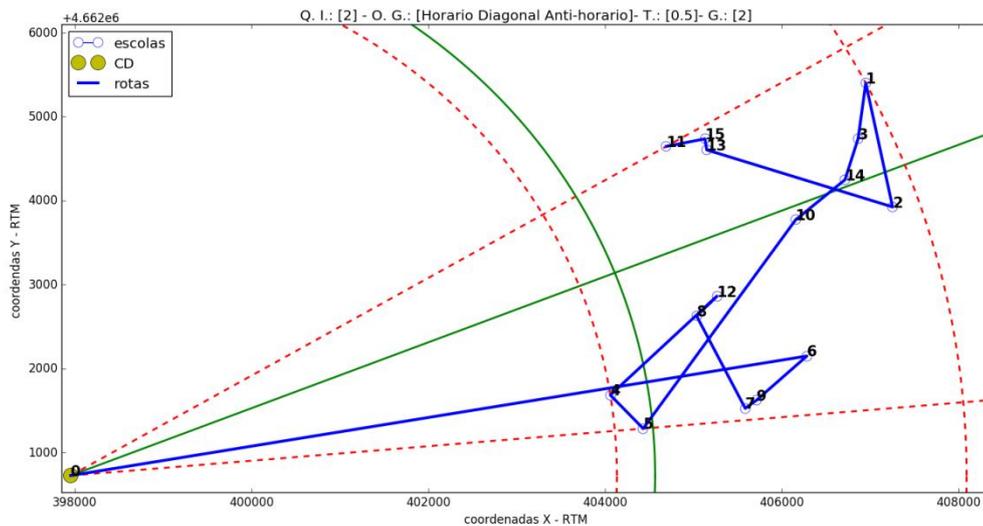


Figura 33 – Solução para $T=0,5$, Geração=2, Quadrante de início =2, Orientação de giro = Horário Diagonal Anti-horário, [0 6 9 7 8 12 4 5 10 14 3 1 2 13 15 11].

Utilizando a **Rotina 1** para encontrarmos o próximo raio e ângulo:
 $m = 0,94, n = 0,93$: Logo, o raio e o ângulo aumentam.

Geração = 3 (vizinho da geração $2/T=0,5$): Figura 34

$$r_{medio} = 7275,11897568 \text{ m}, \theta_{medio} = 0,411014980736 \text{ rad.}$$

Solução = [0 6 9 7 8 12 4 5 10 14 3 1 2 13 15 11].

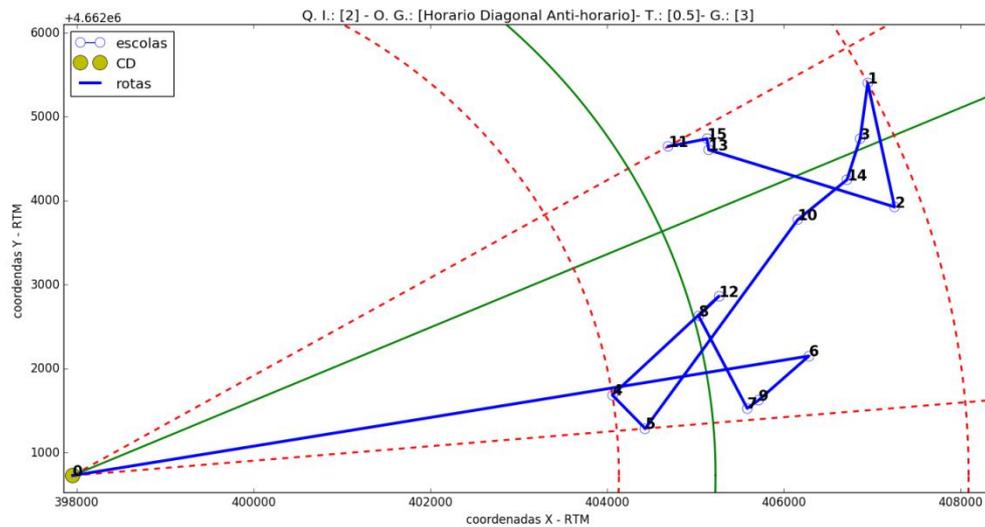


Figura 34 – Solução para $T=0,5$, Quadrante de início = 4, Orientação de giro = Anti-horário, Geração=3, [0 6 9 7 8 12 4 5 10 14 3 1 2 13 15 11].

O exemplo nos mostra como os raios e ângulos escolhidos de forma aleatória alteram os estados vizinhos. Isso indica que o algoritmo tende a gerar estados vizinhos variados. Em comparação com a operação de troca aleatória (*swit*) utilizada em (Lin, Ying, Lee e Chen, 2006), (Mine *et al.*, 2010), (Sakakibara *et al.*, 2010), e, (He *et al.*, 2014), pode-se perceber que a diversidade dos vizinhos gerados por essa nova operação de vizinhança aumenta, pois, ao invés de trocar somente uma escola de posição com outra, esses novos parâmetros podem alterar totalmente a solução inicial quando se trata de um número maior de gerações e temperaturas.

4.2.4 Função objetivo

A função objetivo utilizada para o exemplo da geração de vizinhos por quadrante mostrado anteriormente é a função objetivo 2, que na busca global utiliza a distância euclidiana, e somente a restrição de capacidade. Essa função é utilizada nos métodos GL e QNG. O cálculo dessa função objetivo pode ser visto no trabalho de (Lin *et al.*, 2006).

Em se tratando da função objetivo 1, que é utilizada no método proposto e nos métodos GLG e QNG_GLG, leva em consideração a distância de logradouro (distância

considerando os trajetos das ruas) entre depósito e escola, e de uma escola para outra. Esse cálculo é realizado utilizando-se um algoritmo de otimização para se encontrar a menor distância de logradouro entre os entes anteriormente relacionados. No caso dos métodos GLG e QNG_GLG é utilizado o algoritmo A* para a descoberta dessa distância. Adicionalmente, no método proposto, utiliza-se a heurística do passo seguinte que será vista na próxima seção.

Para a demonstração do cálculo da função objetivo utilizaremos o mesmo exemplo do item anterior, ou seja, 15 escolas e 1 veículo. Adicionalmente, acrescentaremos uma janela de tempo para que seja realizada essa entrega. Essa janela será de 8 horas, o equivalente a 28.800 segundos. Os parâmetros utilizados para a restrição de tempo foram: tempo de deslocamento, tempo de embarque e tempo de desembarque. O tempo de embarque foi definido como 1 hora, o equivalente a 3.600 segundos. O tempo de desembarque foi definido em 30 minutos, o equivalente a 1.800 segundos. O tempo de deslocamento pode ser calculado pela Equação 16. As distâncias calculadas estão em metros, e a velocidade do veículo foi definida em 60 km/h, o equivalente 16,67 m/s aproximadamente.

$$t_{deslocamento} = \frac{\text{distância de deslocamento (m)}}{\text{velocidade do veículo (m/s)}} \quad (16)$$

O tempo total que será atribuído a restrição será o calculado através da Equação 17.

$$t_{total} = t_{embarque} + t_{deslocamento} + t_{desembarque} \quad (17)$$

- 1) Para simplificarmos o exemplo, analisaremos apenas uma temperatura do método SA com 3 gerações. Na geração 1 o cálculo é mais detalhado, enquanto que nas gerações seguintes simplificamos as etapas do cálculo.

1.1) T = 1, Geração = 1 :

1.1.1) Utilizando-se um algoritmo de busca pelo menor caminho calculam-se as distâncias entre as escolas relacionadas na solução inicial e o tempo de deslocamento entre elas. Um exemplo seria a distância percorrida do:

Depósito → Escola 4:

Distância = 19.545,3183665 metros;

tempo de deslocamento = 1.172,7191019903005 segundos;

tempo de embarque = 3.600 segundos;

tempo de desembarque = 1800 segundos;

tempo total = 6.572,7191019903 segundos;

restrição de tempo = 6.572,7191019903 segundos;

restrição de capacidade = 260,277 kg;

Escola 4 → Escola 5:

Distância = 2.518,63200359 metros;

tempo de deslocamento = 151,11792021539998 segundos;

tempo de embarque = 0 segundos;

tempo de desembarque = 1800 segundos;

tempo total = 1.951,1179202153999 segundos;

restrição de tempo = 8.523,83702221 segundos;

restrição de capacidade = 744,729 kg;

1.1.2) As restrições são verificadas a cada percurso. Quando uma das restrições, de tempo ou de capacidade é atingida, é necessário fazer um retorno ao depósito, finalizando assim uma rota. Após o retorno ao depósito, se ainda existirem escolas a serem percorridas, inicia-se uma nova rota. No exemplo, podemos ver algumas restrições que foram

quebradas no decorrer da construção das rotas. Podemos relatar as seguintes restrições: Escola 12 → Escola 6 (restrição de capacidade 2.301 kg), Escola 6 → Escola 13 (restrição de tempo 29.503 segundos), Escola 10 → Escola 14 (restrição de capacidade 2038 kg).

1.1.3) No total foram feitos quatro retornos ao depósito, sendo eles: três restrições e um retorno ao término das entregas de todas as escolas. Por essa razão temos quatro funções objetivo parciais, uma para cada rota, que são:

$$f_1 = 53.227,5684518 \text{ (Rota 1 - Figura 35);}$$

$$f_2 = 62.604,7192443 \text{ (Rota 2 - Figura 36);}$$

$$f_3 = 30.307,139287 \text{ (Rota 3 - Figura 37);}$$

$$f_4 = 25.558,7864328 \text{ (Rota 4 - Figura 38);}$$

O valor total da função objetivo é a soma das funções objetivo parciais, como pode ser visto na Equação 18.

$$f_{total} = f_1 + f_2 + f_3 + f_4 \quad (18)$$

O valor total da função objetivo, nesse caso, é de:

$$f_{total} = 171.698,213416$$

Solução = [0 4 5 7 9 8 12 0 | 0 6 0 | 0 13 15 11 10 0 | 0 14 3 1 2 0]

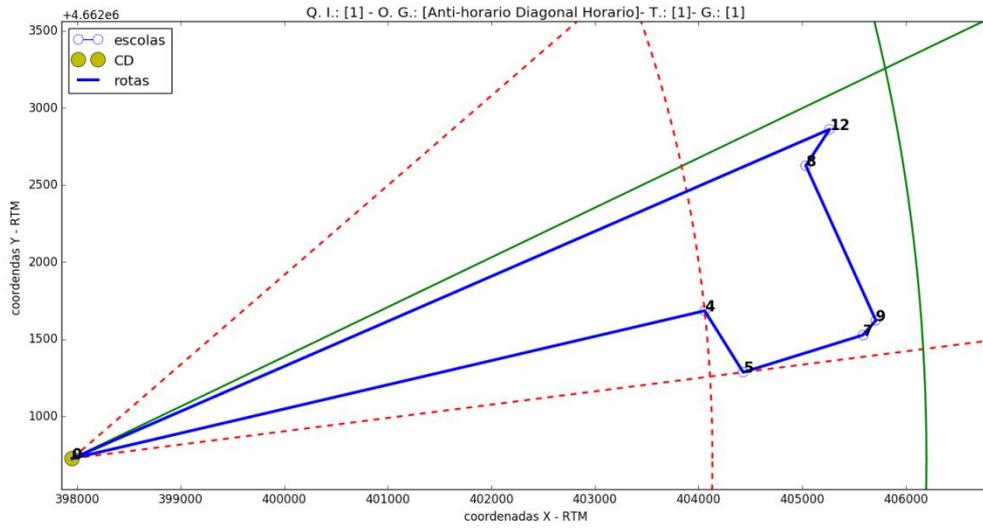


Figura 35 – T=1, Geração=1, Rota1 = [0 4 5 7 9 8 12 0].

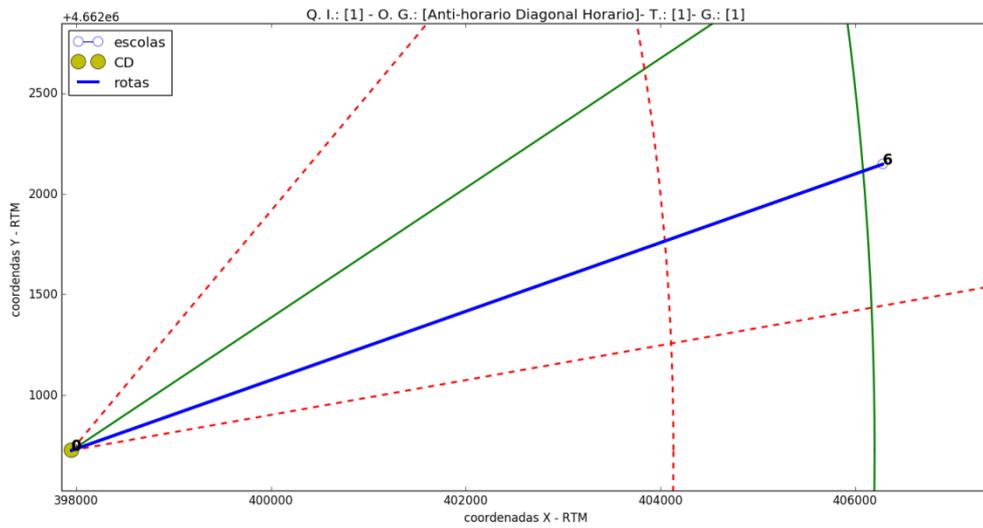


Figura 36 – T=1, Geração=1, Rota2 = [0 6 0].

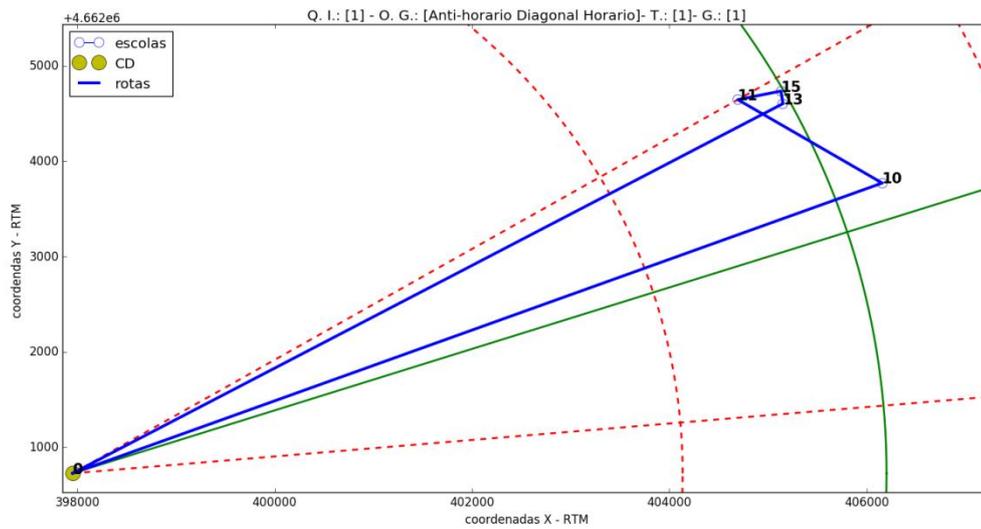


Figura 37 – T=1, Geração=1, Rota3 = [0 13 15 11 10 0].

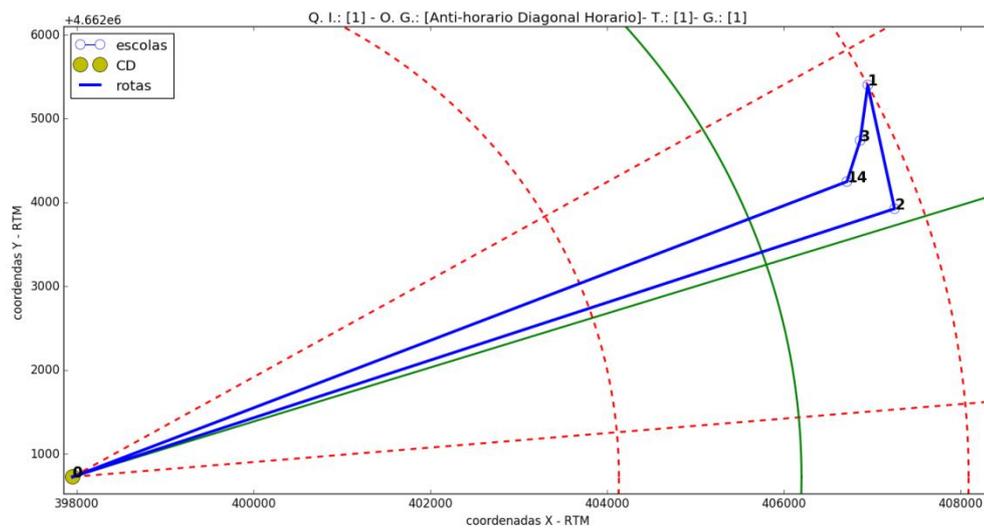


Figura 38 – T=1, Geração=1, Rota4=[0 14 3 1 2 0].

1.2) T = 1, Geração = 2 :

1.2.1) Calculo das distâncias entre as escolas;

1.2.2) As restrições:

Escola 9 → Escola 7 (restrição de capacidade 2.301 kg), Escola 7 →

Escola 10 (restrição de tempo 30.852 segundos), Escola 1 → Escola 2

(restrição de capacidade 2065 kg), Escola 15 → Escola 11 (restrição de tempo 29.110 segundos)

1.2.3) funções objetivo parciais, e total:

$$f_1 = 91.692,6065227 \text{ (Rota 1 - Figura 39);}$$

$$f_2 = 68.547,2192443 \text{ (Rota 2 - Figura 40);}$$

$$f_3 = 28.890,8877826 \text{ (Rota 3 - Figura 41);}$$

$$f_4 = 23.246,8479164 \text{ (Rota 4 - Figura 42);}$$

$$f_5 = 20.441,8192443 \text{ (Rota 5 - Figura 43);}$$

$$f_{total} = 232.819,38071$$

A função objetivo é maior do que a da geração anterior.

Solução = [0 4 5 8 12 6 9 0 | 0 7 0 | 0 10 14 3 1 0 | 0 2 13 15 0 | 0 11 0]

A solução, nesse caso, possui cinco rotas e não quatro como na geração anterior, mostrando que o número de rotas não é fixo.

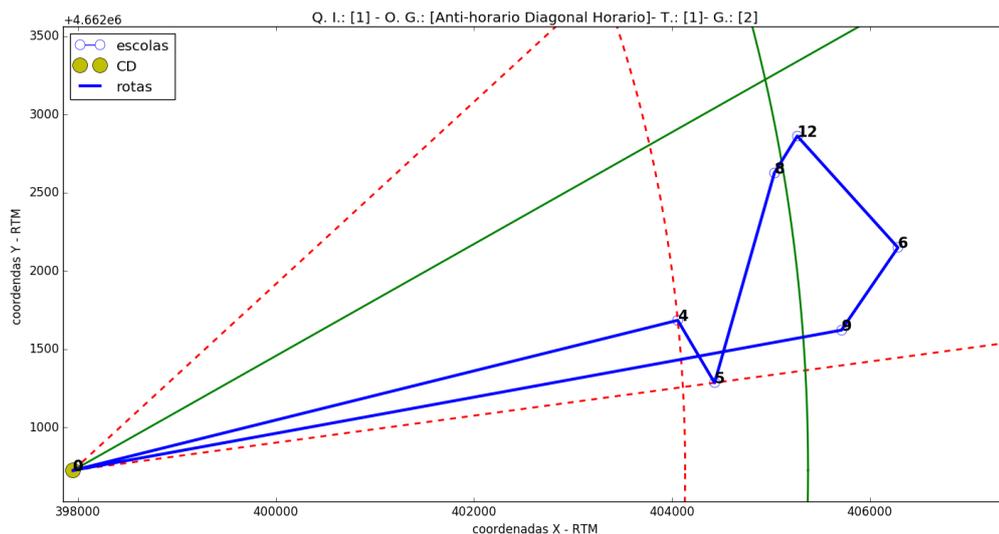


Figura 39 –T=1, Geração=2, Rota1 = [0 4 5 8 12 6 9 0].

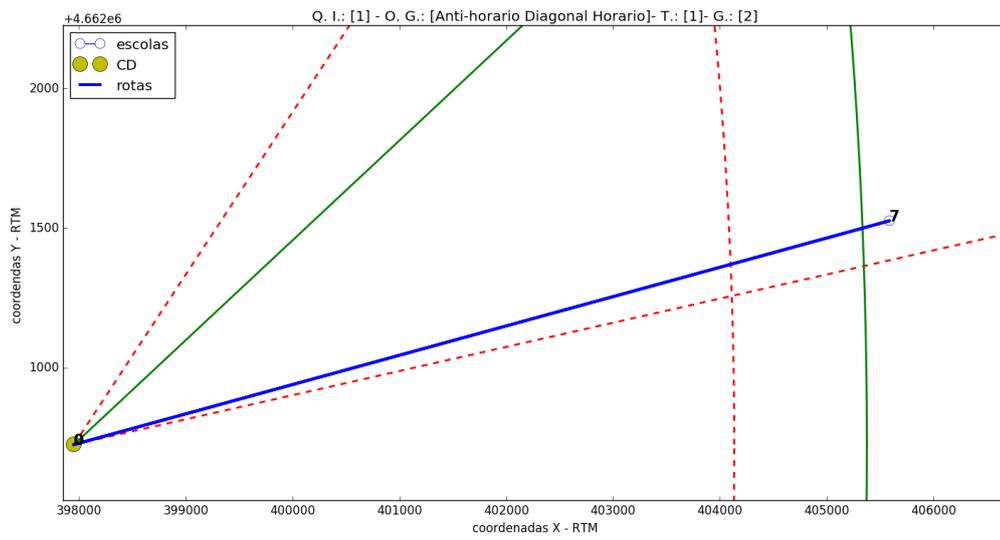


Figura 40 –T=1, Geração=2, Rota2 = [0 7 0].

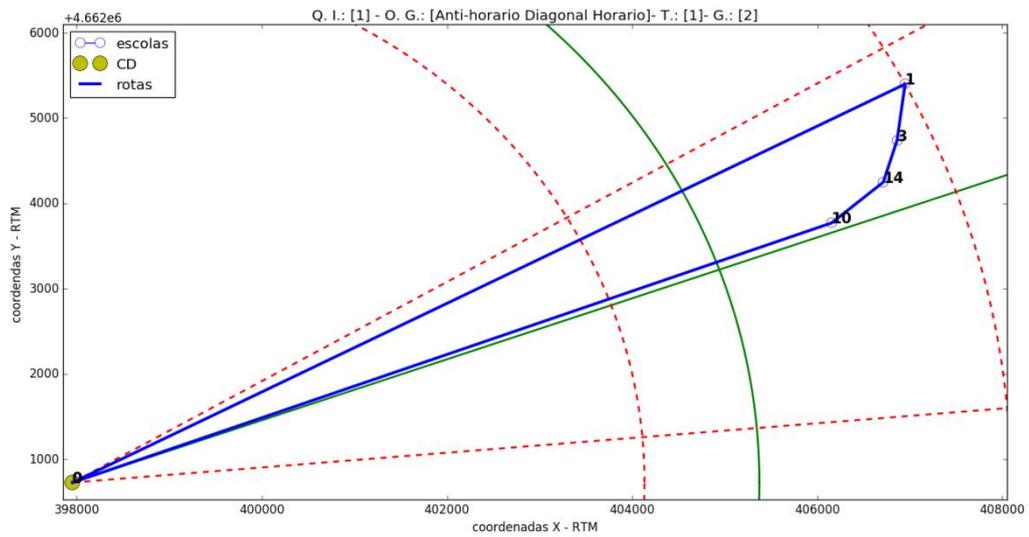


Figura 41 –T=1, Geração=2, Rota3 = [0 10 14 3 1 0].

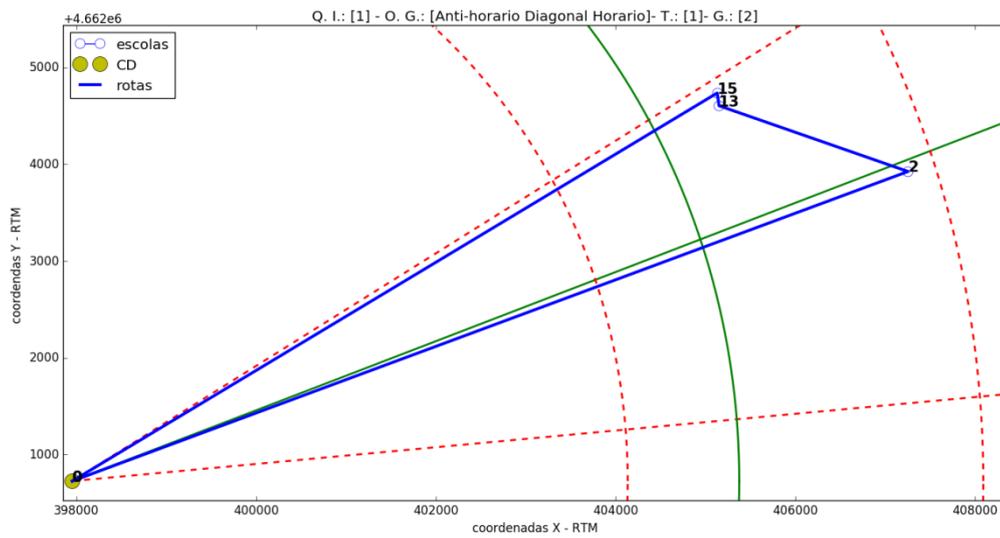


Figura 42 –T=1, Geração=2, Rota4 = [0 2 13 15 0].

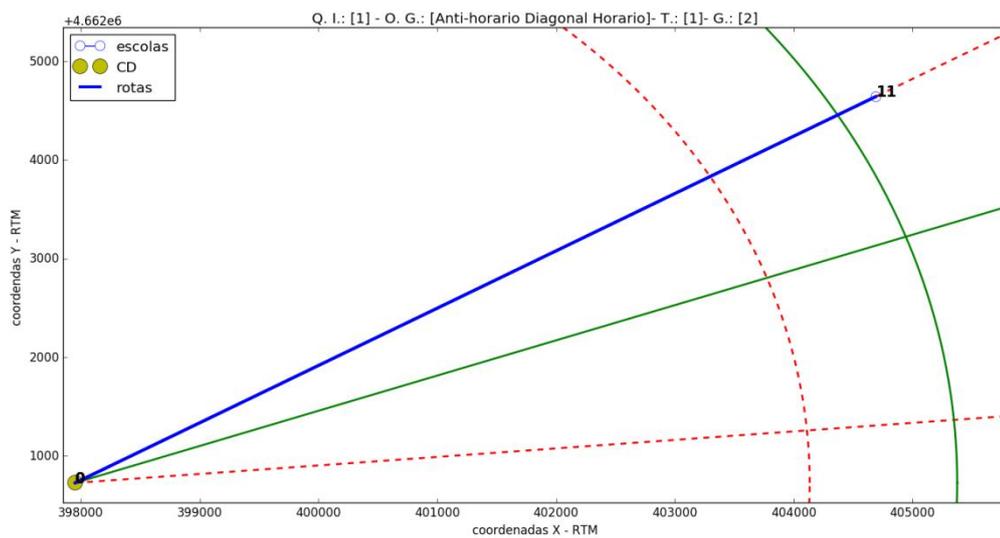


Figura 43 –T=1, Geração=2, Rota5 = [0 11 0].

1.3) T = 1, Geração = 3 :

1.3.1) Calculo das distâncias entre as escolas;

1.3.2) As restrições:

Escola 12 → Escola 6 (restrição de capacidade 2.301 kg), Escola 6 →

Escola 11 (restrição de tempo 29.669 segundos), Escola 10 → Escola 14

(restrição de capacidade 2038 kg).

1.3.3) funções objetivo parciais, e total:

$$f_1 = 53.227,5684518 \text{ (Rota 1 - Figura 44);}$$

$$f_2 = 62.604,7192443 \text{ (Rota 2 - Figura 45);}$$

$$f_3 = 30.846,1978085 \text{ (Rota 3 - Figura 46);}$$

$$f_4 = 25.558,7864328 \text{ (Rota 4 - Figura 47);}$$

$$f_{total} = 172.237,271937$$

A função objetivo é maior do que a da geração anterior.

Solução = [0 4 5 7 9 8 12 0 | 0 6 0 | 0 11 15 13 10 0 | 0 14 3 1 2 0]

A solução, nesse caso, possui cinco rotas.

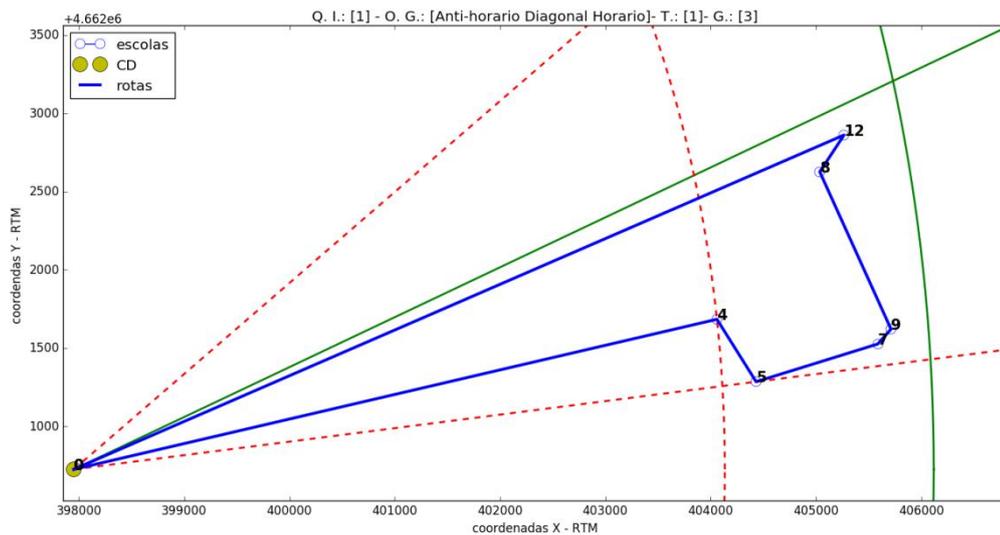


Figura 44 – Solução para T=1, Geração=3, Rota1=[0 4 5 7 9 8 12 0].

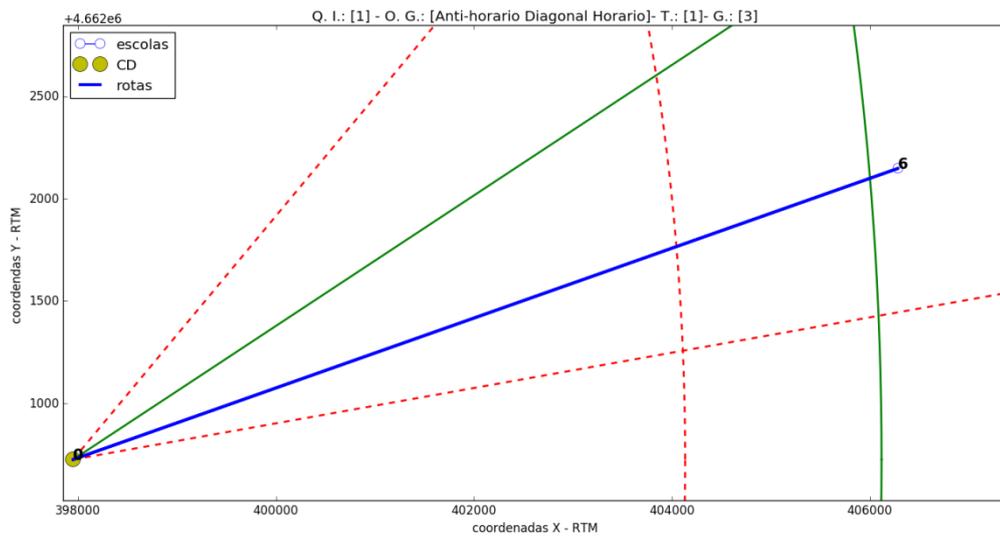


Figura 45 – Solução para T=1, Geração=3, Rota2=[0 6 0].

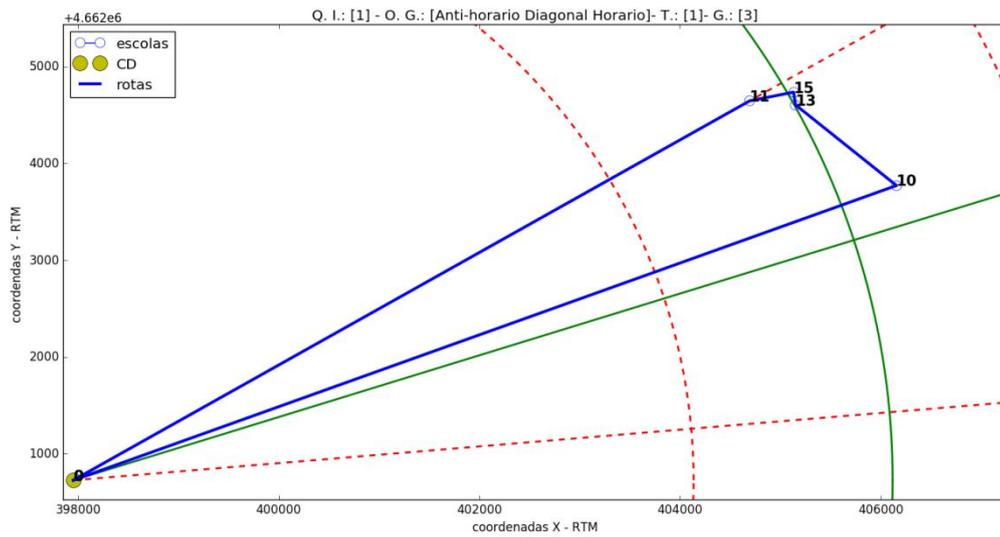


Figura 46 – Solução para T=1, Geração=3, Rota3=[0 11 15 13 10 0].

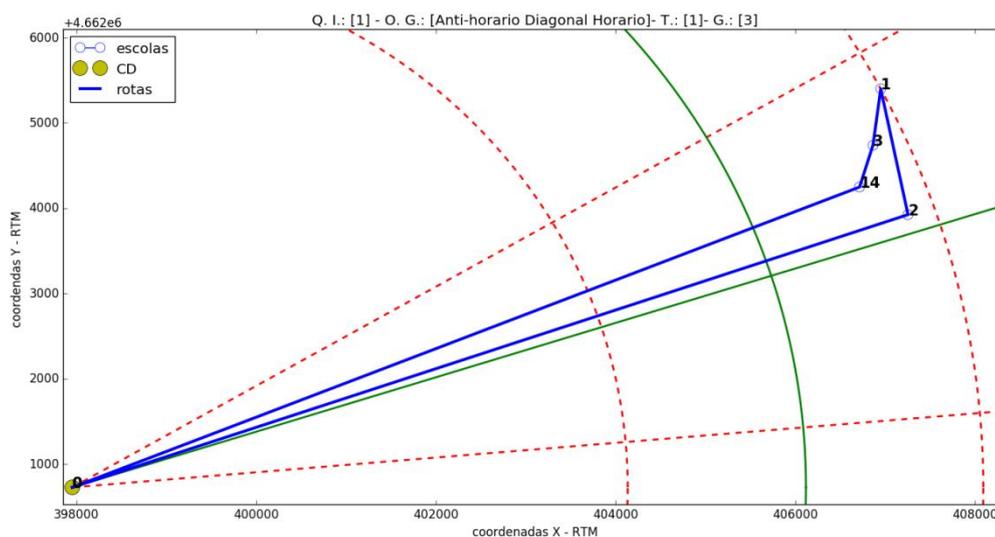


Figura 47 – Solução para T=1, Geração=3, Rota4=[0 14 3 1 2 0].

Quanto aos métodos de aceitação de uma solução, foi utilizado o critério de metrópolis, conforme visto no capítulo anterior, evitando que a solução fique presa em mínimos locais. No caso do exemplo anterior vimos que a melhor solução foi encontrada na geração 3, na temperatura 0,5.

4.2.5 Heurística do passo seguinte

A heurística do passo seguinte tem um conceito simples: seu objetivo é remover logradouros sem saída e quebrar ciclos que ocorrem quando um nodo é alcançado por mais de um percurso diferente, quer seja através de logradouros de mão dupla, através de retornos ou através de viadutos. Para resolver esses dois problemas foi necessária uma análise antecipada dos nodos posteriores e seus vizinhos, antes da expansão dos mesmos. Essa antecipação foi cunhada nesse trabalho de heurística do passo seguinte. A Figura 48 mostra o fluxograma da busca A* com a heurística do passo seguinte.

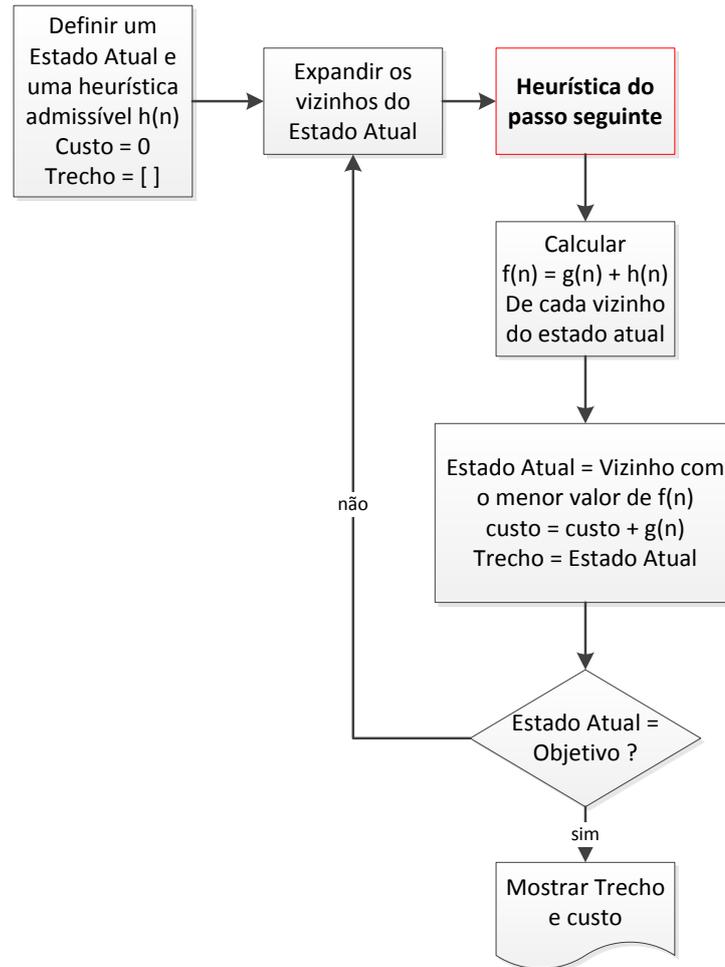


Figura 48 – Fluxograma do algoritmo de busca A* com a heurística do passo seguinte.

A heurística do passo seguinte possui o fluxograma mostrado na Figura 49. A mesma é composta de três blocos consecutivos, que são explicados a seguir.

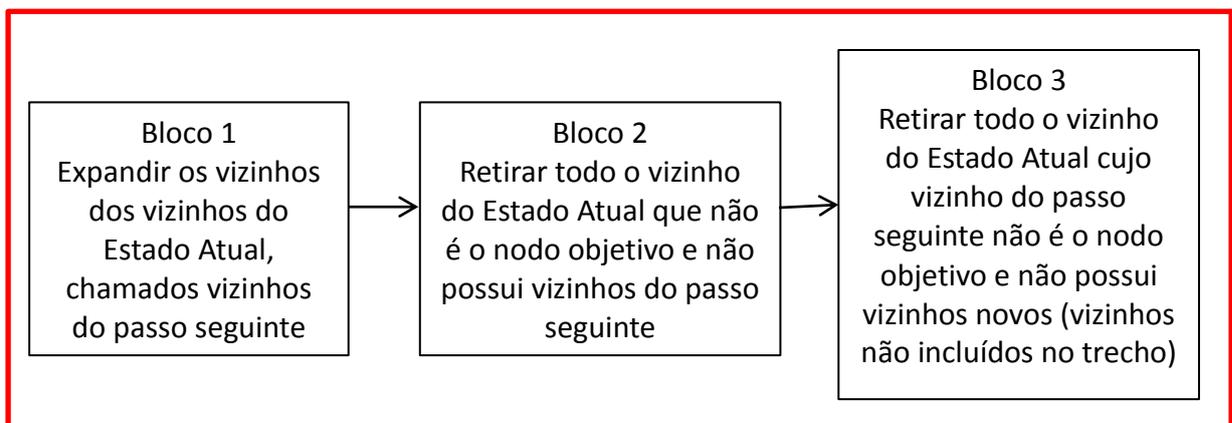


Figura 49 – Fluxograma da heurística do passo seguinte

O fluxograma pode ser melhor entendido se analisarmos o exemplo apresentado na Figura 50. Primeiramente, a proposta é encontrar o menor caminho entre o nodo A e o nodo H. Nesse caso, as setas representam os sentidos dos logradouros.

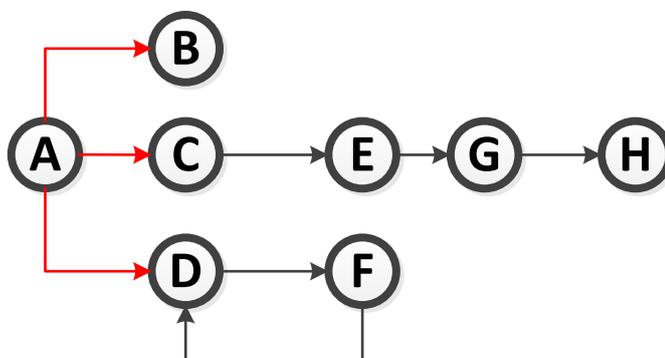


Figura 50 – Exemplo de grafo para a heurística do passo seguinte

Analisando o mapa teremos que o Nodo A possui um total de 3 vizinhos (Nodo B, Nodo C, Nodo D). A heurística do passo seguinte investiga os vizinhos desses vizinhos do Nodo A. Os vizinhos dos vizinhos do Nodo A são chamados de vizinhos do passo seguinte. Na Tabela 6 podemos ver o grau de relacionamento entre os nodos.

Tabela 6 – Vizinhos e vizinhos do passo seguinte no nodo A

Nodo Atual	Vizinhos do nodo atual	Vizinhos dos Vizinhos do nodo atual (vizinhos do passo seguinte)	Vizinhos dos vizinhos do passo seguinte
A	B	-	
	C	E	G
	D	F	D

No Bloco 1 do fluxograma da Figura 48 determinamos os vizinho do nodo atual: B, C e D. No Bloco 2 do fluxograma, expandimos os vizinhos dos nodos B, C e D (vizinhos do passo seguinte): E e F. No Bloco 2 do fluxograma da Figura 48 analisamos quais dos vizinhos

do estado atual, ou seja, B,C e D, não possuem vizinhos do passo seguinte e nem são o nodo objetivo. Após a análise concluímos que o nodo B não possui vizinhos e não é o nodo objetivo. Logo, esse nodo é retirado do grafo, conforme mostrado na Figura 51.

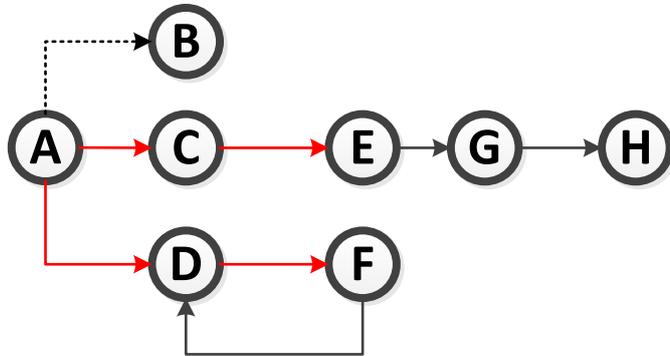


Figura 51 – Exemplo de grafo para a heurística do passo seguinte após a retirada do Nodo B por esse não possuir vizinhos do passo seguinte.

No Bloco 3 do fluxograma da Figura 49 analisamos os vizinhos do passo seguinte, ou seja, E e F, e verificamos que nenhum dos dois é o nodo objetivo, e que o Nodo F não possui vizinhos novos, ou seja seu único vizinho é o nodo D que já está incluído no caminho, causando um ciclo. Logo, o algoritmo remove do grafo o vizinho de A que causará esse ciclo, o nodo D. O resultado após a aplicação da heurística do passo seguinte é mostrado na Figura 52, onde a linha em vermelho é o resultado e as linhas pontilhadas foram retiradas.

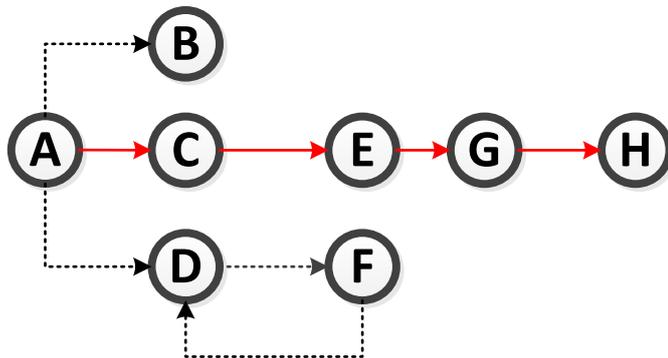


Figura 52 – Grafo final após a aplicação da heurística do passo seguinte.

4.2.6 Plataforma de desenvolvimento

Para realizar as simulações das rotas de entrega de merenda escolar nas escolas municipais de Manaus utilizamos um microcomputador com processador Intel Core i7 @ 3.60 GHz 3.60 GHz e 16 GB de RAM, operando sob a plataforma Microsoft Windows 8. O programa que simula o planejamento das rotas de entrega foi desenvolvido em linguagem Python, assim como os *scripts* que descrevem o funcionamento do algoritmo de geração dos vizinhos, recozimento simulado, Busca A* e algoritmo Dijkstra.

Os materiais relatados neste capítulo foram utilizados nos experimentos que serão apresentados posteriormente. Os métodos descritos e exemplificados mostram as principais contribuições da dissertação ora apresentada. Essas foram: Um novo método de geração de vizinhos por quadrante, um novo método para o cálculo da função objetivo, e a heurística do passo seguinte, utilizada no algoritmo A*. No próximo capítulo apresentamos os resultados dos testes realizados e uma análise dos mesmos.

RESULTADOS E DISCUSSÕES

Este capítulo apresenta os resultados obtidos com o método proposto e o desempenho dos 4 métodos comparativos (Figura 19). O desempenho dos métodos são avaliados em função do tempo de execução e qualidade da solução apresentada, ou seja, menor distância percorrida. Foram realizados 4 experimentos. O experimento 1 visa comparar os métodos GL e GLG. O experimento 2 visa comparar os métodos QNG e QNG_GLG. O experimento 3 visa comparar os métodos SA e QNG. O experimento 4 visa comparar o método proposto e o método QNG_GLG.

5.1 ANÁLISE DE DESEMPENHO DOS MÉTODOS

Para todos os experimentos foram utilizados os seguintes parâmetros para o algoritmo SA: decrescimento de temperatura $c = 0,9$. Isso corresponde a um resfriamento lento, feito com a finalidade de obter um maior número de temperaturas a serem aplicadas, aproximando-se da melhor solução. Após testes realizados foram escolhidos os parâmetros de temperatura (T_i) que melhor se aplicavam 80, 100, 160 e 200. Finalmente, para o número de gerações (n) do experimento 1, que compara os métodos GL e GLG, utilizamos $n = 5, 10, 15$, ou 20. Para os demais experimentos utilizamos $n = 50, 100, 150$ ou 200 com a finalidade de identificar a contribuição do aumento de gerações. A temperatura final (T_f) foi fixada em 1,0.

5.2 EXPERIMENTO 1: COMPARAÇÃO ENTRE OS MÉTODOS GL E GLG

O experimento 1 consiste em testar os métodos GL e GLG com a finalidade de avaliar a nova abordagem de otimização de um VRP global-local-g proposta neste trabalho. O

método GL possui a função objetivo 2, e utiliza a busca global-local. O método GLG possui a função objetivo 1 através da nova abordagem global-local-g proposta neste trabalho. Ambos os métodos usam o operador *swap* para geração de vizinhos. As buscas locais são realizadas com o algoritmo A*. A Tabela 7 mostra o resultado do experimento realizado. As setas azul e vermelha mostram, respectivamente, a relação de diminuição e aumento no Método GLG em comparação ao método GL. As células laranja e vermelha mostram, respectivamente, os melhores e piores resultados.

Tabela 7 – Resultados obtidos com experimento 1.

Temperatura inicial	Número de gerações	Método GL		Método GLG	
		Tempo de simulação (s)	Distância percorrida (m)	Tempo de simulação (s)	Distância percorrida (m)
200	5	725,11	776.691,89	146.144,27	598.217,06
	10	727,58	753.407,85	295.288,63	683.311,61
	15	698,08	724.726,97	443.324,98	661.031,25
	20	660,21	741.270,23	581.757,03	621.775,61
160	5	1.255,69	858.179,35	132.273,83	711.489,18
	10	885,25	803.470,34	268.664,75	667.890,99
	15	972,84	740.822,64	387.218,38	694.707,58
	20	556,80	742.836,33	542.059,34	598.217,06
120	5	974,00	795.632,89	123.674,50	661.031,25
	10	675,95	712.833,53	255.368,82	657.726,42
	15	774,00	750.340,77	373.032,78	517.095,67
	20	1.095,80	821.936,42	498.744,89	635.974,22
80	5	1.037,86	711.489,18	112.444,99	598.217,06
	10	827,03	850.038,30	242.663,73	621.775,61
	15	1.114,15	840.878,93	341.334,55	694.707,58
	20	669,05	667.890,99	442.322,66	618.316,45

A Tabela 7 mostra que o método GLG apresenta tempos de simulação maiores do que o método GL. Acredita-se que a principal razão é que no método GL, a busca local A* é executada apenas uma vez, após todas as buscas globais. Enquanto que, no método GLG, a busca local é aplicada um número de vezes igual ao produto entre: o número de gerações em cada temperatura, o número de valores de temperaturas usados no recozimento simulado e o número médio *g* de *clusters* em que uma solução global é dividida. Para a distância

percorrida, o método GLG apresenta valores menores que o método GL. As principais razões para este comportamento são as seguintes: no método GLG a busca local é aplicada a cada iteração da busca global, enquanto que no método GL é a busca local é aplicada uma única vez; A função objetivo 1 no método GLG é calculada usando a distância de logradouro. Enquanto que para o cálculo da função objetivo 2 do método GL é empregada a distância euclidiana para otimização global. A menor distância percorrida encontrada no método GLG foi de 517.095,67 metros e a pior foi de 711.489,18, enquanto no método GL, a menor distância percorrida foi de 667.890,99 metros e a maior de 858.179,35. Pode-se ver que o método que possui a função objetivo 1, método global-local-g, teve melhores resultados do que o método GL, que possui a função objetivo 2, método global-local.

5.3 EXPERIMENTO 2: COMPARAÇÃO ENTRE OS MÉTODOS GL E QNG

O experimento 2 consiste em testar os métodos GL e QNG com a finalidade, primeiramente, de avaliar um aumento do número de gerações no método GL. No experimento 1 foram apresentados os resultados do método GL com 5, 10, 15, e 20 gerações. No experimento 2 foram apresentados os resultados do mesmo método com 50, 100, 150, e 200 gerações. Depois de avaliarmos o método GL isoladamente, então podemos avaliar o novo método de geração de vizinhos apresentado neste trabalho, comparando os métodos GL e QNG com o número de gerações para ambos os métodos de 50, 100, 150, e 200. Os métodos GL e QNG utilizam a busca global-local. A busca local final é realizada através do algoritmo A*. A única diferença entre os métodos é a técnica de geração de vizinhos. O método GL utiliza o operador *swap* para geração de vizinhos, enquanto que o método QNG utiliza a geração de vizinhos por quadrante. A Tabela 8 mostra os resultados obtidos. As setas verdes mostram a relação de melhoria nos resultados do Método GL do experimento 2 em relação ao mesmo método no experimento 1. As setas azul e vermelha, respectivamente,

mostram a relação de diminuição e aumento no QNG em comparação ao método GL. As células laranja e vermelha mostram, respectivamente, os melhores e piores resultados.

Tabela 8 – Resultados obtidos com experimento 2.

Temperatura inicial	Número de gerações	Método GL		Método QNG	
		Tempo de simulação (s) ↓	Distância percorrida (m) ↓	Tempo de simulação (s) ↑	Distância percorrida (m) ↓
200	50	569,28	722.043,48	4.427,61	344.482,74
	100	502,58	753.407,85	6.765,89	250.525,14
	150	431,33	683.311,61	10.861,66	238.758,89
	200	472,26	635.974,22	18.633,84	280.026,07
160	50	395,70	621.775,61	3.344,46	225.982,23
	100	885,25	803.470,34	6.617,59	318.286,20
	150	410,04	618.316,45	13.249,33	258.421,40
	200	556,80	742.836,33	14.416,90	252.143,11
120	50	339,57	517.095,67	3.318,16	208.006,32
	100	472,71	712.833,53	6.377,25	277.848,48
	150	545,61	750.340,77	9.581,10	267.476,69
	200	639,16	792.279,82	13.635,61	284.415,28
80	50	564,43	711.489,18	3.017,95	242.665,91
	100	559,96	721.098,35	7.579,12	344.482,74
	150	548,80	708.399,41	9.227,50	344.482,74
	200	540,73	716.443,50	11.145,46	262.857,32

A Tabela 8 mostra que o método GL quando aplicado um número maior de gerações na ordem de grandeza de 10 vezes, em comparação com o mesmo método apresentado no experimento anterior, Tabela 7, em via de regra possui uma melhoria quanto a distância percorrida. Sendo que, dentro de uma mesma tabela nada se pode afirmar quanto ao efeito do aumento do número de gerações. Na Tabela 8 alguns resultados são melhores como quando se analisa a temperatura 160 com 100 gerações que teve como resultado 803.470 metros e depois do aumento para 150 gerações o resultado passa a ser 618.316 metros. Porém, em outros casos são piores como na temperatura 120 com 50 gerações teve como resultado 517.095, e com o aumento do número de gerações para 100 obteve resultado pior de 712.833 metros. Isso se deve a solução inicial que é escolhida de forma aleatória, podendo essa contribuir ou não para a convergência até a solução ótima. O método QNG apresenta tempos de simulação

maiores do que o método GL. Acredita-se que a principal razão é a aplicação do método de geração de vizinhos por quadrante, por essa ser a única diferença entre os mesmos. Para a distância percorrida, o método QNG apresenta valores menores que o método GL, tendo como a principal razão a aplicação do método de geração de vizinhos por quadrante. Logo, pode-se afirmar que a método de geração de vizinhos por quadrante melhorou a métrica de distância mínima em aproximadamente 500 km, com na temperatura 160 e número de gerações 100.

5.4 EXPERIMENTO 3: COMPARAÇÃO ENTRE OS MÉTODOS QNG E QNG_GLG

O experimento 3 consiste em testar os métodos QNG e QNG_GLG com a mesma finalidade do experimento 1, avaliar a nova abordagem de otimização de um *VRP* global-local-g proposta neste trabalho. Nesse experimento, os números de gerações utilizadas foram 50, 100, 150, e 200. O método de geração de vizinhos utilizado é o método proposto neste trabalho, chamado de geração de vizinhos por quadrante. O método QNG utiliza a busca global-local. O método QNG_GLG utiliza a nova abordagem global-local-g apresentada neste trabalho. A busca local é realizada pelo algoritmo A* (Tabela 9). Onde, As setas azul e vermelha, respectivamente, mostram a relação de diminuição e aumento no QNG_GLG em comparação ao método QNG. As células laranja e vermelha mostram, respectivamente, os melhores e piores resultados. A busca local também é realizada pelo algoritmo Dijkstra (Tabela 10). Onde, As setas vermelhas mostram a relação de aumento no QNG com Dijkstra em comparação ao método QNG com A*. As células laranja e vermelha mostram, respectivamente, o melhor e o pior resultado.

Tabela 9 – Resultados obtidos com experimento 3 - busca local: Algoritmo A*.

Temperatura inicial	Número de gerações	Método QNG		Método QNG_GLG	
		Tempo de simulação (s)	Distância percorrida (m)	Tempo de simulação (s) ↑	Distância percorrida (m) ↓
200	50	4.427,61	344.482,74	345.145,57	207.693,76
	100	6.765,89	250.525,14	810.661,04	214.483,37
	150	10.861,66	238.758,89	1.122.155,55	214.204,85
	200	18.633,84	280.026,07	1.496.207,40	207.693,76
160	50	3.344,46	225.982,23	334.883,15	207.693,76
	100	6.617,59	318.286,20	718.766,35	214.483,66
	150	13.249,33	258.421,40	1.056.194,52	207.693,76
	200	14.416,90	252.143,11	1.437.482,70	207.693,76
120	50	3.318,16	208.006,32	337.380,47	214.483,37
	100	6.377,25	277.848,48	663.570,57	214.483,37
	150	9.581,10	267.476,69	994.574,58	208.006,32
	200	13.635,61	284.415,28	1.284.542,22	214.483,37
80	50	3.017,95	242.665,91	333.801,61	222.563,06
	100	7.579,12	344.482,74	644.356,67	214.204,71
	150	9.227,50	344.482,74	982.007,55	214.483,37
	200	11.145,46	262.857,32	1.127.573,60	208.006,32

A Tabela 9 mostra que o método QNG_GLG apresentou tempos de simulação maiores do que o método QNG. Acredita-se que a principal razão é a mesma apresentada no experimento 1. Para a distância percorrida, o método QNG_GLG apresenta valores menores que o método QNG, tendo como as principais razões as mesmas do experimento 1. Entretanto, os resultados apresentados neste experimento tiveram uma melhoria significativa quanto aos apresentados no experimento 1. As duas modificações feitas para este experimento foram: um aumento no número de gerações e uma mudança no método de geração de vizinhos. Logo, pode-se afirmar que o método QNG foi melhor que os métodos comparativos SA e GLG. Entretanto, o método QNG_GLG é o melhor a ser utilizado até o presente momento, por esse ter encontrado a menor distância percorrida de 207.693,76 metros.

Tabela 10 – Resultados obtidos com experimento 3 - busca local: Algoritmo Dijkstra.

Temperatura inicial	Número de gerações	Método QNG		Método QNG_GLG	
		Tempo de simulação (s)	Distância percorrida (m)	Tempo de simulação (s)	Distância percorrida (m)
200	50	3.889,11	1.185.244,03	T.O.	-
	100	9.375,64	1.185.244,03	T.O.	-
	150	11.667,33	1.069.151,52	T.O.	-
	200	15.556,43	1.185.244,03	T.O.	-
160	50	5.147,71	1.069.151,52	T.O.	-
	100	7.751,82	1.088.574,88	T.O.	-
	150	12.512,24	1.088.574,88	T.O.	-
	200	15.002,56	1.185.244,03	T.O.	-
120	50	4.066,62	1.069.151,52	T.O.	-
	100	9.656,86	1.185.244,03	T.O.	-
	150	12.523,47	1.122.347,22	T.O.	-
	200	19.330,17	1.069.151,52	T.O.	-
80	50	3.537,14	1.253.662,32	T.O.	-
	100	6.426,58	1.122.347,22	T.O.	-
	150	9.644,49	1.088.574,88	T.O.	-
	200	14.852,42	1.088.574,88	T.O.	-

Na Tabela 10 realizamos o mesmo experimento com o algoritmo Dijkstra. Essa tabela mostra que o método QNG_GLG, utilizando o método Dijkstra, não obteve resultado até o limite do tempo (TIME OUT – T.O.) de 20 dias, 1.728.000 segundos, ou seja, a busca local com o algoritmo de Dijkstra não obteve resultado, mostrando maior tempo de execução do que o algoritmo A* apresentado na Tabela 9. O método QNG obteve resultados dentro do tempo limite. Entretanto, os resultados apresentados para a métrica de distância percorrida foram altos em comparação com o experimento mostrado na Tabela 9, para o algoritmo A*. Conclui-se, então, que não é viável a utilização do algoritmo Dijkstra na busca local do mapa apresentado.

5.5 EXPERIMENTO 4: COMPARAÇÃO ENTRE O MÉTODO QNG_GLG E O MÉTODO PROPOSTO

O experimento 4 consiste em testar o método QNG_GLG e o método proposto, com a finalidade de avaliar o desempenho da heurística do passo seguinte, proposta nesse trabalho para acelerar o método de busca local A*. No experimento 3 foram apresentados os resultados do método QNG_GLG. Esse método utiliza duas das contribuições propostas no trabalho. A única contribuição que esse método não utiliza é a heurística do passo seguinte, aplicada no algoritmo de busca local A*. O método proposto faz uso de todas as contribuições deste trabalho: a nova abordagem global-local-g caracterizada pela função objetivo 1, com distância de logradouro, o método de geração de vizinhos por quadrante, e a heurística do passo seguinte aplicada na busca local. A Tabela 11 mostra os resultados obtidos.

Tabela 11 – Resultados obtidos com experimento 4.

Temperatura inicial	Número de gerações	Método QNG_GLG		Método Proposto	
		Tempo de simulação (s)	Distância percorrida (m)	Tempo de simulação (s)	Distância percorrida (m)
200	50	345.145,57	207.693,76	75.911,54	214.483,66
	100	810.661,04	214.483,37	149.714,14	207.693,76
	150	1.122.155,55	214.204,85	222.822,68	207.693,76
	200	1.496.207,40	207.693,76	278.360,70	207.693,76
160	50	334.883,15	207.693,76	66.022,93	214.204,85
	100	718.766,35	214.483,66	132.740,28	207.693,76
	150	1.056.194,52	207.693,76	236.000,68	207.693,76
	200	1.437.482,70	207.693,76	309.016,60	207.693,76
120	50	337.380,47	214.483,37	59.074,29	214.483,66
	100	663.570,57	214.483,37	128.699,38	207.693,76
	150	994.574,58	208.006,32	298.429,05	207.693,76
	200	1.284.542,22	214.483,37	294.213,31	207.693,76
80	50	333.801,61	222.563,06	46.252,59	207.693,76
	100	644.356,67	214.204,71	102.058,57	214.483,66
	150	982.007,55	214.483,37	219.854,00	207.693,76
	200	1.127.573,60	208.006,32	269.746,02	214.204,85

A Tabela 11 mostra que o método 4 apresenta tempos de simulação maiores do que o método proposto. Acredita-se que a principal razão é a aplicação da heurística do passo seguinte no algoritmo A* utilizado na busca local. Para a distância percorrida, o método 4 e o método proposto apresentam valores muito próximos, a menor distância encontrada em todos os experimentos. É relevante afirmar-se, no entanto, que a heurística do passo seguinte reduziu o tempo de simulação em até 14 dias, com a temperatura inicial 200 e número de gerações 200.

CONCLUSÕES

Nesse trabalho propôs-se um novo método para solução do problema VRP, com significativas contribuições. O estudo de caso que serviu para avaliação das contribuições propostas foi a atividade de entrega de merenda escolar na rede pública de ensino da cidade de Manaus, em uma sub-região contendo 54 escolas. As principais métricas avaliadas foram a distância percorrida na rota e o tempo de simulação para se obter uma solução.

A primeira contribuição foi uma nova abordagem global-local-g, onde uma sequência repete-se por inúmeras iterações: candidato a solução global, definição de “g” agrupamentos grupos e realização de “g” buscas locais cada uma busca global com “g”. Na busca global utilizou-se a meta-heurística do recozimento simulado. Essa meta-heurística foi também utilizada em alguns dos trabalhos revisados (Lin, Ying, Lee e Hsi, 2006), (Lin, Ying, Lee e Chen, 2006), e, (Sakakibara *et al.*, 2010). Diferentemente desses trabalhos, no trabalho ora apresentado a busca local foi aplicada após cada resultado intermediário da busca global. Esse foi um grande diferencial do método proposto nessa dissertação, que permitiu obter valores significativamente menores para a métrica distância percorrida na rota. Nos outros trabalhos revisados, as distâncias entre as escolas são calculadas utilizando-se a distância euclidiana e a busca local é aplicada apenas após o término da busca global, como nos métodos 1 e 3 apresentados no capítulo anterior, e cujos resultados foram bem inferiores em termos da métrica distância percorrida da rota.

Na busca local o algoritmo de busca A* apresentou resultados muito superiores ao algoritmo Dijkstra, tanto em termos da métrica distância percorrida da rota quanto em termos da métrica tempo de simulação.

A modelagem realizada no trabalho, considerando a distância de logradouro, permite ao motorista não apenas saber a sequência em que as escolas devem ser percorridas, mas também o caminho para se percorrer as mesmas. A distância real percorrida no caminho de uma escola a outra foi nomeada de distância de logradouro, e, é através dessa distância que é calculada a função objetivo otimizada.

A segunda importante contribuição deste trabalho foi a forma de gerar soluções vizinhas no algoritmo SA. Neste trabalho foi introduzida uma nova forma de geração de vizinhos, chamada de geração de vizinhos por quadrante. A geração de vizinhos por quadrante mostrou-se uma maneira eficaz para obtenção de melhores resultados em termos da métrica distância percorrida na rota, em comparação com a métrica de swap, conforme pode ser visto nos resultados do experimento 3. O método swap é muito utilizado em outros trabalhos revisados da literatura (Lin, Ying, Lee e Chen, 2006), (Mine *et al.*, 2010) e (Sakakibara *et al.*, 2010), e, (He *et al.*, 2014).

A última contribuição foi a implementação de uma nova heurística na busca local, a heurística do passo seguinte. Essa heurística foi aplicada no algoritmo de busca A* com a finalidade de melhorar o tempo de resposta do algoritmo sem perder a qualidade na métrica distância percorrida na rota. A associação da heurística do passo seguinte ao algoritmo de busca A* representou um ganho significativo de tempo de simulação, conforme pode ser observado através dos resultados do experimento 4.

O trabalho teve algumas limitações, alguns dos nodos tiveram que ser retirados, pois não utilizamos o mapa da cidade como um todo, somente a região IV. Os algoritmos utilizados para encontrar o menor caminho tiveram um tempo considerável em algumas situações, ou seja dependeu da solução inicial gerada.

Para trabalhos futuros sugerimos aplicar os métodos desenvolvidos nesse trabalho em benchmarks da área. Além disso, a utilização de novas meta-heurísticas e buscas locais podem ser testados para melhorar o tempo de execução do algoritmo. A utilização de um monitoramento de tráfego pode ser inserido para aproximar o problema ainda mais da realidade.

REFERÊNCIAS

BAI, Y.; ZHOU, X.; ZHANG, Y.; YANG, M. **Study of multi-vehicle routing problem with time window**, 12^a International Symposium on Operations Research and Its Applications (ISORA), Luoyang, China, p.21-28, 2015.

BALINSKI, M. L.; QUANDT, R. E. **On an integer program for a delivery problem**. Operations Research, v. 12, n. 2, p. 300-304, 1964. ISSN 0030-364X.

CHEN, M. H.; CHANG, P. C.; CHIU, C. Y.; ANNADURAI, S. P. **A hybrid two-stage sweep algorithm for capacitated vehicle routing problem**. International Conference on Control, Automation and Robotics (ICCAR), p.195-199, 2015.

CLARKE, G. U.; WRIGHT, J. W. **Scheduling of vehicles from a central depot to a number of delivery points**. Operations research, v. 12, n. 4, p. 568-581, 1964. ISSN 0030-364X.

CORMEN, T. H. **Introduction to algorithms**. MIT press, 2009. ISBN 0262533057.

DANTZIG, G. B.; RAMSER, J. H. **The Truck Dispatching Problem**. Management Science, v. 6, n. 1, p. 80-91, 1959/10/01 1959. ISSN 0025-1909.

DIJKSTRA, E. W. **A note on two problems in connexion with graphs**. Numerische mathematik, v. 1, n. 1, p. 269-271, 1959. ISSN 0029-599X.

DORIGO, M.; GAMBARDELLA, L. M. **Ant colony system: a cooperative learning approach to the traveling salesman problem**. IEEE Transactions on Evolutionary Computation, v. 1, n. 1, p. 53-66, 1997. ISSN 1089-778X.

DUAN, F.; FU, Z. **Research on the Vehicle Routing Problem with Pickup and Delivery Soft Time Windows and Its Tabu Search Algorithm**. Computer Engineering & Science, v. 3, p. 021, 2009.

ERDOĞAN, S.; MILLER-HOOKS, E. **A Green Vehicle Routing Problem**. Transportation Research Part E: Logistics and Transportation Review, v. 48, n. 1, p. 100-114, 2012. ISSN 1366-5545.

FANGGENG, Z.; DONG, M.; JIANGSHENG, S.; WEIMIN, L. **A hybrid genetic algorithm for the vehicle routing problem with simultaneous pickup and delivery**. Chinese Control and Decision Conference, p. 3928-3933, 2009.

FISCHETTI, M.; GONZÁLEZ, J. J. S.; TOTH, P. **A branch-and-cut algorithm for the symmetric generalized traveling salesman problem**. Operations Research, v. 45, n. 3, p. 378-394, 1997.

GLOVER, F. **Heuristics for integer programming using surrogate constraints**. Decision Sciences, v. 8, n. 1, p. 156-166, 1977. ISSN 1540-5915.

_____. **Future paths for integer programming and links to artificial intelligence.** Computers & operations research, v. 13, n. 5, p. 533-549, 1986. ISSN 0305-0548.

GOLDEN, B. L.; WASIL, E. A.; KELLY, J. P.; CHAO, I. M. **The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results.** In: (Ed.). Fleet management and logistics: Springer, p.33-56, 1998.

GOMORY, R. **An algorithm for the mixed integer problem.** RAND CORP SANTA MONICA CA, 1960.

GUAN, C. H.; CAO, Y.; SHI, J. **Tabu Search Algorithm for Solving the Vehicle Routing Problem.** Third International Symposium on Information Processing (ISIP), p.74-77, 2010.

HAI, S.; YUNLONG, Z.; LI, J.; WENPING, Z. **Two-phase heuristic for Capacitated Vehicle Routing Problem.** Second World Congress on Nature and Biologically Inspired Computing (NaBIC), p.534-539, 2010.

HE, Y.; MIAO, W.; XIE, R.; SHI, Y. **A tabu search algorithm with variable cluster grouping for multi-depot vehicle routing problem.** 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Proceedings of the IEEE, p.12-17, 2014.

HOLLAND, J. H. **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.** U Michigan Press, 1975. ISBN 0472084607

Instituto de Brasileiro de Geografia e Estatística. **Senso Demográfico 2000/2010:** IBGE 2010.

_____. **Arranjos Populacionais e Concentrações Urbanas do Brasil.** GEOCIÊNCIAS, D. D. e GEOGRAFIA, C. D. Rio de Janeiro - RJ: IBGE 2015.

JINHUI, G.; YING, X.; HONGZHEN, W. **An Improved TS for the Open Vehicle Routing Problem with Soft Time Windows.** Fifth International Symposium on Computational Intelligence and Design (ISCID), p.382-385, 2012.

KIM, G.; ONG, Y. S.; HENG, C. K.; TAN, P. S.; ZHANG, N. A. **City Vehicle Routing Problem (City VRP): A Review.** IEEE Transactions on Intelligent Transportation Systems, v. 16, n. 4, p. 1654-1666, 2015. ISSN 1524-9050.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. **Optimization by simulated annealing.** science, v. 220, n. 4598, p. 671-680, 1983. ISSN 0036-8075.

KRAMER, R. H. F. R.; SUBRAMANIAN, A.; PENNA, P. H. V. **Problema de roteamento de veículos assimétrico com frota heterogênea limitada: um estudo de caso em uma indústria de bebidas.** Gestão & Produção, v. 23, p. 165-176, 2016. ISSN 0104-530X.

LAND, A. H.; DOIG, A. G. **An automatic method of solving discrete programming problems.** Econometrica: Journal of the Econometric Society, p. 497-520, 1960. ISSN 0012-9682.

LAPORTE, G.; GENDREAU, M.; POTVIN, J. Y.; SEMET, F. **Classical and modern heuristics for the vehicle routing problem**. International Transactions in Operational Research, v. 7, n. 4–5, p. 285-300, 2000. ISSN 0969-6016.

LIMA, M.; LOBO, A. **Custos Logísticos no Brasil**. 2014. Disponível em: < <http://www.ilos.com.br/web/custos-logisticos-no-brasil/> >. Acesso em: 25 de março de 2016.

LIN, S. **Computer solutions of the traveling salesman problem**. The Bell System Technical Journal, v. 44, n. 10, p. 2245-2269, 1965. ISSN 0005-8580.

LIN, S. W.; YING, K. C.; LEE, Z. J.; CHEN, H. S. **Vehicle Routing Problems with Time Windows Using Simulated Annealing**. IEEE International Conference on Systems, Man and Cybernetics. SMC '06., p.645-650, 2006.

_____. **Applying Simulated Annealing Approach for Capacitated Vehicle Routing Problems**. IEEE International Conference on Systems, Man and Cybernetics, SMC '06., p.639-644, 2006.

MARQUES, D. D. S.; ARAÚJO, C. S. D.; RODRIGUES, H. F. Roteirização do transporte da merenda escolar das escolas municipais urbanas de Manaus usando SIG. XX Simpósio de Engenharia de Produção - SIMPEP, 2013.

MILIOTIS, P. **Integer programming approaches to the travelling salesman problem**. Mathematical Programming, v. 10, n. 1, p. 367-378, 1976. ISSN 0025-5610.

MINE, M. T.; SILVA, M. S. A.; OCHI, L. S.; SOOUZA, M. J. F.; SILVA, T. C. B. **O problema de roteamento de veículos com coleta e entrega simultânea: uma abordagem via Iterated Local Search e GENIUS**. 2010. ISSN 2237-1346

MLADENOVIC, N. **A variable neighborhood algorithm-a new metaheuristic for combinatorial optimization**. p.112, 1995.

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. ISSN 0160-5682.

MONTEMANNI, R.; GAMBARDELLA, L.; RIZZOLI, A.; DONATI, A. **A new algorithm for a dynamic vehicle routing problem based on ant colony system**. Second international workshop on freight transportation and logistics, p.27-30, 2013.

PADBERG, M.; RINALDI, G. **A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems**. SIAM review, v. 33, n. 1, p. 60-100, 1991. ISSN 0036-1445.

PILLAC, V.; GENDREAU, M.; GUÉRET, C.; MEDAGLIA, A. L. **A review of dynamic vehicle routing problems**. European Journal of Operational Research, v. 225, n. 1, p. 1-11, 2013. ISSN 0377-2217.

POP, P.; CHIRA, C. **A hybrid approach based on genetic algorithms for solving the Clustered Vehicle Routing Problem**. IEEE Congress on Evolutionary Computation (CEC), p.1421-1426, 2014.

REPOUSSIS, P. P.; TARANTILIS, C. D.; IOANNOU, G. **The Open Vehicle Routing Problem with Time Windows**. The Journal of the Operational Research Society, v. 58, n. 3, p. 355-367, 2007. ISSN 01605682, 14769360.

RUSSEL, S.; NORVIG, P. **Inteligência artificial**. Editora Campus, p. 26, 2004.

SAKAKIBARA, K.; TSUDA, T.; NISHIKAWA, I. **Simulated annealing method based on recursive problem decomposition for vehicle routing problems**. Proceedings of SICE Annual Conference, p.1016-1020, 2010.

SANTOS, E.; OCHI, L. S.; SIMONETTI, L.; GONZÁLEZ, P. H. **A Hybrid Heuristic based on Iterated Local Search for Multivehicle Inventory Routing Problem**. Electronic Notes in Discrete Mathematics, v. 52, p. 197-204, 6// 2016. ISSN 1571-0653.

SKOK, M.; SKRLEC, D.; KRAJCAR, S. **The genetic algorithm method for multiple depot capacitated vehicle routing problem solving**. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies, p.520-526 vol.2, 2000.

SOLOMON, M. M. **Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints**. Operations Research, v. 35, n. 2, p. 254-265, 1987. ISSN 0030364X, 15265463.

SOUZA, S. C. B. D. **Planejamento de trajetória para um robô móvel com duas rodas utilizando um algoritmo A-Estrela modificado**. 2008.

SUBRAMANIAN, A.; UCHOA, E.; OCHI, L. S. **A hybrid algorithm for a class of vehicle routing problems**. Computers & Operations Research, v. 40, n. 10, p. 2519-2531, 2013. ISSN 0305-0548.

TANIGUCHI, E. **City Logistics: Network Modelling and Intelligent Transport Systems**. Elsevier Science, 2001. ISBN 9780080439037.

TIWARI, A.; CHANG, P.-C. **A block recombination approach to solve green vehicle routing problem**. International Journal of Production Economics, v. 164, p. 379-387, 2015.

TUNJONGSIRIGUL, B.; PONGCHAIRERKS, P. **A Genetic algorithm for a vehicle routing problem on a real application of Bakery delivery**. International Conference on Electronic Computer Technology (ICECT), p.214-217, 2010.

UCHOA, E.; CALDAS, M.; LIMA, F. Q.; DRUMMOND, L. **Algoritmos para problemas reais de roteamento de veículos: uma análise comparativa**. XXXVIII Simpósio Brasileiro de Pesquisa Operacional, 2006.

WHITLEY, L. D. **The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best**, p.116-123, 1989.

WREN, A.; CARR, J. D. **Computers in transport planning and operation.** 1971.

YEN, J.; LIAO, J. C.; LEE, B.; RANDOLPH, D. **A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method.** IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), v. 28, n. 2, p. 173-191, 1998. ISSN 1083-4419.

ZHIBIN, Y.; QIANWEN, G.; QIANG, X. **City logistics transportation invisible cost control based on ITS.** 6th Advanced Forum on Transportation of China (AFTC 2010), p.204-210, 2010.