



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
PÓS-GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

LEANDRO CESAR RIBEIRO

ALGORITMO EVOLUTIVO COM METODOLOGIA HÍBRIDA
PARA ELABORAÇÃO DE ROTAS DE DISTRIBUIÇÃO

UBERLÂNDIA

2019

LEANDRO CESAR RIBEIRO

ALGORITMO EVOLUTIVO COM METODOLOGIA HÍBRIDA PARA
ELABORAÇÃO DE ROTAS DE DISTRIBUIÇÃO

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, como exigência parcial para obtenção do Título de Mestre em Ciências.

Área de Concentração: Sistemas de Energia Elétrica.

Linha de Pesquisa: Controle e Automação

Orientador: Prof. Dr. Aniel Silva de Moraes.

UBERLÂNDIA

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

R354a Ribeiro, Leandro Cesar, 1977-
2019 Algoritmo evolutivo com metodologia híbrida para elaboração de
rotas de distribuição [recurso eletrônico] / Leandro Cesar Ribeiro. -
2019.

Orientador: Aniel Silva de Moraes
Dissertação (mestrado) - Universidade Federal de Uberlândia,
Programa de Pós-Graduação em Engenharia Elétrica.
Modo de acesso: Internet.
Disponível em: <http://dx.doi.org/10.14393/ufu.di.2019.985>
Inclui bibliografia.
Inclui ilustrações.

1. Engenharia elétrica. 2. Programação genética (Computação). 3.
Algoritmo de Dijkstra. 4. Empresas - Automação. I. Moraes, Aniel Silva
de, 1979-, (Orient.) II. Universidade Federal de Uberlândia. Programa de
Pós-Graduação em Engenharia Elétrica. III. Título.

CDU:621.3

Glória Aparecida – CRB-6/2047

2019
LEANDRO CESAR RIBEIRO

ALGORITMO EVOLUTIVO COM METODOLOGIA HÍBRIDA PARA
ELABORAÇÃO DE ROTAS DE DISTRIBUIÇÃO

Dissertação aprovada para obtenção do título de
Mestre em Ciências no Programa de Pós-
Graduação em Engenharia Elétrica da
Universidade Federal de Uberlândia (MG) pela
banca examinadora formada por:

Uberlândia, 15 de Abril de 2019.

Prof. Dr. Aniel Silva de Moraes, UFU/MG

Prof. Dr. Josué Silva de Moraes, UFU/MG

Prof. Dr. Fábio Vincenzi Romualdo da Silva, UFU/MG

Prof. Dr. Kleber Lopes Fontoura, CEFET/MG

Dedico este trabalho a Deus e a minha esposa
Manuela, pelo estímulo, carinho, paciência e
compreensão.

RESUMO

Na cadeia logística de entrega de encomendas de um modo geral, são notáveis as dificuldades de se elaborar as rotas de distribuição. O presente trabalho tem como objetivo o desenvolvimento de um protótipo de software capaz de otimizar a rota de distribuição de encomendas, utilizando como estudo de caso as encomendas registradas dos correios na cidade de Coromandel/MG. Destaca-se neste trabalho a opção de elaboração de rotas que facilitam a sua utilização de acordo com o tipo de veículo utilizado pelo entregador das encomendas. Foi utilizada a base de dados OpenStreetMap para localização das coordenadas geográficas das residências e ruas, sendo que as rotas são definidas através de um algoritmo genético híbrido baseado no problema do caixeiro viajante e utilizando o algoritmo de Dijkstra para definir qual o caminho de custo mínimo dentre as vértices do problema. Neste trabalho também são descritas as etapas de elaboração, bem como os resultados obtidos de forma experimental. Ao final, conclui-se através deste trabalho que a utilização de algoritmos híbridos apresenta um grande potencial para a resolução de problemas de roteamento, incluindo aqueles nos quais são considerados o tipo de veículo a ser utilizado nas entregas.

Palavras Chaves: Algoritmo de Dijkstra. Algoritmos Genéticos Híbridos. Correios de Coromandel. Problema do Caixeiro Viajante. Roteirização.

ABSTRACT

In the logistical chain of parcel delivery in general, the difficulties of managing the routes of distribution are remarkable. The present work has the objective of developing a software prototype capable of optimizing the order distribution route, using as a case study the registered orders of the post office in the city of Coromandel / MG. It is highlighted in this paper the option of elaborating routes that facilitate their use according to the type of vehicle used by the delivery of orders. An OpenStreetMap database was used to locate the geographical coordinates of residences and streets, and the routes are defined through a hybrid genetic algorithm based on travelling salesman problem and also with the Dijkstra algorithm to define the minimum cost path between the vertices of the problem. This work is also described as elaboration stages, as well as the results obtained experimentally. Finally, it is concluded through this work that the use of hybrid algorithms presents a great potential for the resolution of routing problems, including those in which the type of vehicle to be used in the deliveries is considered.

Key Words: Dijkstra Algorithm. Hybrid Genetic Algorithms. Coromandel Post Office. Traveling Salesman Problem. Scripting.

LISTA DE ILUSTRAÇÕES

Figura 1 - Interface Gráfica Openstreetmap	14
Figura 2 - PCV Simétrico	15
Figura 3 - PCV Assimétrico	16
Figura 4 - Representação de um grafo	18
Figura 5 - Fluxograma do Algoritmo de Dijkstra.....	19
Figura 6 - Grafo Ilustrativo.....	20
Figura 7 - Matriz de Vínculos e Pesos.....	20
Figura 8 - Grafo Ilustrativo com o caminho mínimo calculado	22
Figura 9 - Fluxograma básico do sistema de entregas proposto.....	27
Figura 10 - Vista por satélite da ECT de Coromandel-MG.....	28
Figura 11 - Parte da matriz esparsa que representa os pontos de interligação das ruas	29
Figura 12 - Fluxograma do pré-processamento do banco de dados	30
Figura 13 - Pontos de Interligação da Agência dos Correios	31
Figura 14 - Matriz Esparsa Modificada: Conexões do Ponto 374.....	32
Figura 15 - Representação de um cromossomo válido.....	33
Figura 16 - Representação gráfica do cruzamento PMX.....	34
Figura 17 - Representação gráfica da Mutação Swap	35
Figura 18 - Avaliação das distâncias entre P1 e P2	37
Figura 19 - Distâncias entre P1 e P2.....	37
Figura 20 – Diagrama de Voronoi.....	39
Figura 21 – Esboço do Mapa da Rota.....	39
Figura 22 – Seleção do Ponto de Passagem	40
Figura 23 – Detalhe do Roteamento	40
Figura 24 – Cruzamento sem sinal	41
Figura 25 – Conversão à direita ou à Esquerda	42
Figura 26 – Rede de estradas dependente apenas da distância.....	43
Figura 27 – Rede de estradas dependente da distância e das condições do tráfego	43
Figura 28 - Sistema Carro Q moto	44
Figura 29 - Pontos de entrega de encomendas – S1	47
Figura 30 - Percurso mínimo estipulado pelo AG – S1.....	48
Figura 31 - Cromossomo vencedor	49

Figura 32 - Rota de Entrega – S1	49
Figura 33 - Percurso mínimo estipulado pelo AG para Veículos – S1.....	50
Figura 34 - Cromossomo vencedor S1 - Rota para veículos	51
Figura 35 - Rota de entrega para veículos – S1	51
Figura 36- Pontos de entrega de encomendas - S2.....	52
Figura 37 - Percurso mínimo estipulado pelo AG – S2.....	53
Figura 38 - Cromossomo vencedor – S2	53
Figura 39 - Rota de Entrega - AG sem restrição S2	54
Figura 40 - Percurso mínimo estipulado pelo AG para Veículos – S2.....	55
Figura 41 - Cromossomo vencedor S2 - Rota para veículos	55
Figura 42 - Rota de entrega para veículos – S2.....	56
Figura 43 - Resultado intermediário estipulado pelo AG – S2.....	57
Figura 44 - Cromossomo vencedor – Resultado Intermediário.....	57
Figura 45 - Rota de Entrega Intermediária	58
Figura 46 - Rota Alternativa segundo informações on-line do trânsito	59

LISTA DE TABELAS

Tabela 1 - Produção científica ('traveling salesman problems' or 'routeing' or 'routing')	11
Tabela 2 - Produção científica ((<i>'traveling salesman problems' or 'routeing' or 'routing'</i>) <i>and 'Genetic Algorithm'</i>).....	11
Tabela 3 - Equipamento utilizado na simulação.....	46

SUMÁRIO

1 INTRODUÇÃO	8
1.1 Objetivos e Contribuição da Dissertação.....	10
1.2 Motivação	10
1.3 Organização do Trabalho	12
2 MATERIAIS E MÉTODOS.....	13
2.1 Descrição do local do estudo/problema	13
2.2 Base de Dados OpenStreetMap.....	13
2.3 Problema do Caixeiro Viajante.....	14
2.4 Algoritmo de Dijkstra	18
2.5 Algoritmos Genéticos Híbridos	22
2.5.1 Representação do Cromossomo.....	23
2.5.2 Operadores Genéticos.....	24
2.5.3 Avaliação	25
3 METODOLOGIA.....	27
3.1 OpenStreetMap.....	28
3.2 Desenvolvimento do Algoritmo Genético Híbrido.....	32
3.2.1 Arquitetura do Cromossomo	32
3.2.2 Operador de Seleção.....	33
3.2.3 Operador de Cruzamento.....	34
3.2.4 Operador de Mutação	35
3.2.5 Avaliação	36
4 TRABALHOS RELACIONADOS	39
4.1 Algoritmos Híbridos.....	39
4.2 Algoritmo de Dijkstra como critério de avaliação.....	40
4.3 Ajustes na Matriz de Vínculos e Pesos.....	41

4.4	Informações sobre o Trânsito no Cálculo das Rotas	42
4.5	Elaboração de Rotas Considerando o Veículo Utilizado	44
5	RESULTADO E DISCUSSÃO.....	46
5.1	Simulação 1	47
5.2	Simulação 2	52
5.3	Sugestão de Melhoria	59
6	CONCLUSÕES.....	60
	REFERÊNCIAS	62

1 INTRODUÇÃO

Atraindo pesquisadores de diferentes campos, entre os quais pesquisa operacional, matemática, física, biologia, inteligência artificial, o Problema do Caixeiro Viajante – PCV (do inglês *Traveling Salesman Problem* - TSP) é um dos mais proeminentes dentre um amplo conjunto de problemas de otimização combinatória (LAPORTE, 1992). Apesar de apresentar uma simplicidade na sua formulação, no PCV é possível envolver otimização combinatória da forma mais ampla possível; motivo pelo qual ele é muito usado como “*benchmark*” para avaliação de novos algoritmos e ou estratégias de solução (REINELT, 1994).

Problemas de produção que consistem em minimizar o tempo total de execução de várias tarefas em um conjunto de equipamentos, de cristalográfica, de controle de robôs e de otimização de percurso seja nas linhas de montagem de componentes eletrônicos ou na definição da melhor logística de entrega para empresas varejistas vislumbra-se o emprego de algoritmos idealizados para resolver o PCV (HELSGAUN, 2000). Assim como questões relacionadas ao desempenho computacional tem atraído o interesse por ele, inúmeros problemas reais também são modelados como problema do tipo caixeiro viajante ou suas variantes.

Mas um dos problemas que se dá muito destaque ao PCV são os problemas de roteirização de veículos, esse termo vem do inglês “*routing*” (ou “*routeing*”) designando a determinação de uma ou mais paradas a serem seguidas por veículo de uma frota, no intuito de visitar um conjunto de pontos geograficamente dispersos, com o menor percurso possível e ou menor tempo possível. O termo roteamento de veículo também é utilizado alternativamente por alguns autores (CUNHA, 1997). Esse tipo de problemas é muitas vezes definido como um problema de um ou mais caixeiros viajantes que podem incluir ou não restrições, tais como por capacidade de carga, dirigibilidade e tamanho do veículo, fluxo apresentados pelas vias de acesso, etc.

Quando se tem um conjunto de cidades (ou consumidores), cada um com uma demanda q_i por um produto, e um depósito com veículos de capacidade Q , e é necessário encontrar a rota dos veículos minimizando os custos de transporte nós temos o Problema de Roteamento de Veículos (PRV). Como uma generalização do Problema do Caixeiro Viajante (PCV), o PRV pertence à classe de problemas NP-difíceis (do inglês “*NP-hard*”) (LENSTRA, 1981), o que significa que possuem ordem de complexidade exponencial e

um algoritmo em tempo polinomial para encontrar a solução ótima deste tipo de problema não é conhecido. Existem algoritmos exatos que raramente resolvem instâncias envolvendo mais que 50 cidades (RENAUD & BOCTOR, 2002). Em suma, isto significa que os métodos aplicados a instancias reais de PRV são, em geral, heurísticos, isto é, não asseguram a obtenção ótima do ponto de vista matemático.

Porém os problemas de PRV estão tão presentes num país como o Brasil que de acordo com a Confederação Nacional do Transporte (CNT), no ano de 2017, o país possuía cerca de 212.866 km de rodovias pavimentadas e 1.365.426 km de não pavimentadas ligando os 5.570 municípios brasileiros distribuídos ao longo de uma área territorial de 8.515.767,049 km². Ainda de acordo com a CNT, a distribuição espacial da logística de transportes no país apresentava uma predominância por rodovias, chegando a totalizar 61% de toda a carga transportada. (CNT, 2017).

Além das dificuldades em se elaborar as rotas de distribuição, estabelecer o roteamento dos veículos e na capacitação dos profissionais de entrega, o transporte é a área que mais eleva os custos operacionais, chegando à marca de um a dois terços do valor total da cadeia logística (VIEIRA, 2013).

Com o aumento da utilização da internet e maior confiança nos comércios digitais, ocorreu um acréscimo significativo no volume e diversidade das compras efetuadas através das plataformas online e, conseqüentemente uma maior demanda logística de distribuição (ESTADÃO CONTEÚDO, 2017).

Comumente conhecido como Correios, a Empresa de Correios e Telégrafos (ECT) é a empresa responsável pela maior parte dos serviços postais do Brasil. Chegando a distribuir milhões de encomendas diariamente em todos os municípios do território brasileiro (CORREIOS, 2016).

De acordo com o senso dos Correios, no mês de abril de 2016 chegaram a ser distribuídos aproximadamente 33,2 milhões de objetos por dia através de uma frota de 25.236 veículos (motocicletas, veículos leves e pesados) e de suas 10.524 unidades de distribuição espalhas pelo país. Só no ano de 2015 foram entregues cerca de 8,3 bilhões de encomendas (CORREIOS, 2016). A logística por trás destes serviços se torna complexa e muito cara. Ao todo, cerca de 11,6% do Produto Interno Bruto Brasileiro (PIB) é destinado aos gastos logísticos (MAIA e BERNARDES, 2016).

Neste trabalho é proposto um método híbrido, no qual técnicas heurísticas, como o algoritmo genético são combinadas com um método exato como o Algoritmo de Dijkstra,

para resolver o PRV. Inicialmente uma solução é gerada aleatoriamente e coordenadas geográficas para esta solução são adquiridas através da base de dados *OpenStreetMap*. Porém como diferencial, este trabalho se baseia numa proposta focada em restrições, ou seja, o algoritmo proposto avalia a rota também tendo como meta verificar a dirigibilidade e tamanho do veículo assim como o fluxo apresentados pelas vias de acesso.

Também será apresentado um estudo de caso com dados dos Correios da cidade de Coromandel/MG, embasando a viabilidade da metodologia proposta, haja visto que esta unidade dos correios dispõe de um fluxo de aproximadamente 200 objetos registrados a cada dia, necessitando de entrega imediata dentro dos liames de restrições discutidos acima.

1.1 Objetivos e Contribuição da Dissertação

Este trabalho tem como principal objetivo desenvolver e avaliar uma metodologia capaz de prover uma rota para um problema de PRV, porém agregando algumas restrições que os casos reais dispõem e são pouco avaliados. Ou seja, otimizar além do tempo gasto e o espaço percorrido para a execução da rota ele também visará atribuir apenas rotas que prezem pela dirigibilidade do veículo, assim como com menor fluxo dos mesmos.

A topologia foi capaz de encontrar soluções para problemas de rotas, que é um problema “*NP-hard*”, apresentando um custo computacional viável quando restrições na solução são inseridas, o que aumenta expressivamente seu custo.

Isto somente foi possível com uma proposta que concatena algoritmos aproveitando características de destaque de cada um.

1.2 Motivação

Devido aos grandes avanços em áreas que demandam em alguma etapa de sua concepção por algoritmos de rota, observou-se que estes algoritmos também sofreram um aumento considerável de pesquisas nas últimas décadas, constatada através de uma pesquisa de revisão bibliográfica, realizada em Janeiro de 2019. Nesta pesquisa foi selecionado todas as bases de dados disponíveis pelo mecanismo de busca do “IEEE Xplore Digital Library” e configurado como critério de filtro as palavras chaves: ‘traveling salesman problems’ or ‘routeing’ or ‘routing’ encontrando 455.841 resultados entre

Artigos, Jornais, Revistas, Atas de Conferências e etc. no período compreendido entre 1872 e 2019 apresentados na Tabela 1.

Tabela 1 - Produção científica ('traveling salesman problems' or 'routeing' or 'routing')

Ano	Resultados	
	Unidade	Percentual
Até 1990	35959	7,89%
1990 - 1999	54863	12,04%
2000 - 2009	145223	31,86%
2010 - 2019	219796	48,22%
Total	455841	

Fonte: Adaptado de (IEEE Xplore Digital Library, 2019).

Nesta mesma linha de raciocínio, utilizando as palavras chaves: (('traveling salesman problems' or 'routeing' or 'routing') and 'Genetic Algorithm') foram encontrados 22.375 resultados, conforme apresenta a tabela 2, observa-se que existe um crescimento do uso de algoritmos genéticos em solução de problemas de roteamento, mas que ele não é o algoritmo predominante neste tipo de estudo haja visto que na última década 219.796 trabalhos foram desenvolvidos na temática e destes 13.599 são utilizando algoritmo genético.

Tabela 2 - Produção científica (('traveling salesman problems' or 'routeing' or 'routing') and 'Genetic Algorithm')

Ano	Resultados	
	Unidade	Percentual
Até 1990	96	0,43%
1990 - 1999	1724	7,71%
2000 - 2009	6956	31,09%
2010 - 2019	13599	60,78%
Total	22375	

Fonte: Adaptado de (IEEE Xplore Digital Library, 2019).

No intuito por identificar quais algoritmos que se sobressaem no contexto de rotas, duas pesquisas chamaram a atenção, quando utilizado como palavra-chave (('traveling salesman problems' or 'routeing' or 'routing') and 'Dijkstra') identificando 682 trabalhos e uma pesquisa com (('salesman problems' or 'routing' or 'routeing') and 'Manhattan')

encontrando 354 trabalhos. O fato é que ambos são algoritmos para cálculo de menor distância, mas para este contexto aparente não apresentam bons resultados comparados a outros algoritmos. Quando optou-se por pesquisar ((*'salesman problems'* or *'routing'* or *'routeing'*)) and (*'Genetic Algorithm'* and *'Dijkstra'*)) apenas 33 trabalhos foram encontrados e o primeiro foi no ano de 1999. Verificou-se assim duas possibilidades: baixa capacidade de ambos trabalharem juntos ou pouco foi estudado sobre este tipo de uso. Avaliando estas pesquisas identificou-se que nenhuma das mesmas, utilizava do AG e o Dijkstra da forma aqui proposta.

Então optou-se por analisar esta topologia e resultados promissores foram identificados.

1.3 Organização do Trabalho

A organização do trabalho é dada da seguinte forma:

- Capítulo 2: Apresenta alguns conceitos importantes para proporcionar um embasamento teórico, necessário para o entendimento da proposta deste trabalho.
- Capítulo 3: Fundamentalmente expõem de forma detalhada, as técnicas que foram aplicadas e as que foram desenvolvidas no decorrer da execução do projeto.
- Capítulo 4: Compara trabalhos já publicados relacionados com os assuntos e técnicas envolvidas na elaboração do projeto
- Capítulo 5: Apresenta os resultados reais obtidos com a utilização do código desenvolvido, realiza uma análise preliminar e discute os resultados obtidos.
- Capítulo 6: Conclui o trabalho com uma discussão final dos resultados, apresenta o conhecimento adquirido e nos sugere pontos que podem ser aperfeiçoados em estudos futuros.

2 MATERIAIS E MÉTODOS

Nesta seção são discutidos conceitos importantes para o entendimento da proposta apresentada neste trabalho.

2.1 Descrição do local do estudo/problema

Diariamente, ao meio dia chegam ao município de Coromandel/MG milhares de objetos classificados como simples e registrados. Dentre os simples, incluem-se: as cartas e outras encomendas que não necessitam de assinatura. Já em relação aos objetos registrados, o município recebe cerca de 200 SEDEX e PAC's dos quais precisam ser entregues aos destinatários espalhados dentre aproximadamente 120 ruas da cidade. As tentativas de entrega dos objetos registrados têm por obrigação serem efetuadas no mesmo dia ao qual chegam no município. Ao desembarcar nas mediações dos correios é destinado a um dos carteiros a separação das encomendas. O mesmo fica responsável por elaborar a rota do veículo de entrega. Logo após a ordenação dos objetos é passado o leitor de código de barras para o *check-in* dos objetos no sistema. Após este processo é gerado uma lista de assinaturas de recebimento, local onde o cliente irá confirmar que recebeu a encomenda. Esta lista também serve como uma espécie de guia para o entregador, uma vez que ela contém em ordem os produtos e seus respectivos endereços de entrega. No exemplo em questão é o próprio criador da lista que é o responsável pela distribuição dos produtos aos destinatários finais.

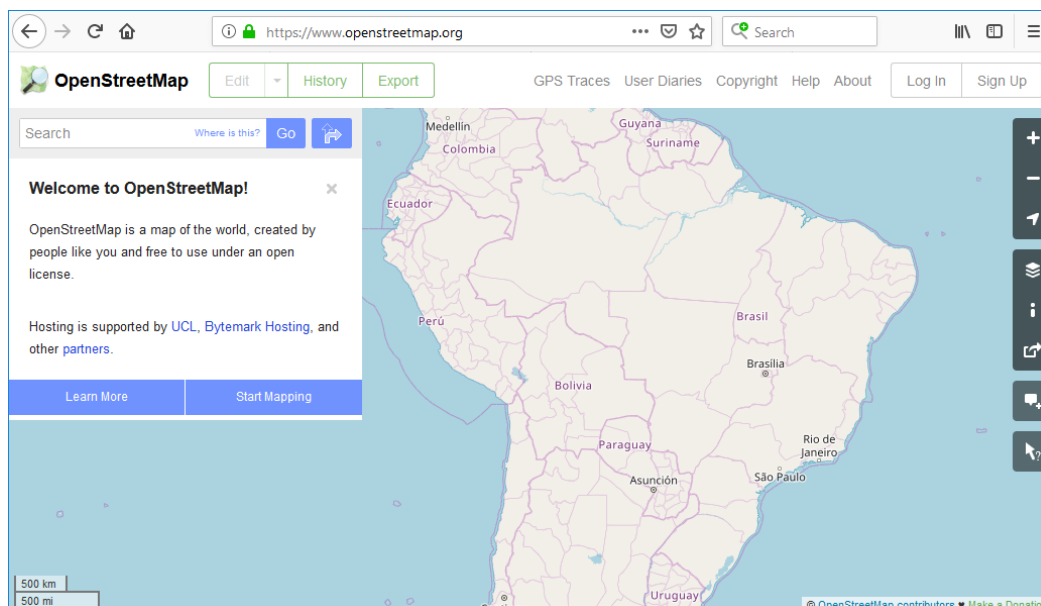
A elaboração da rota é totalmente dependente de um profissional que conheça a cidade ou os setores de distribuição (em cidades maiores) e saiba ordenar as encomendas a fim de criar um mapa de distribuição dos produtos.

2.2 Base de Dados OpenStreetMap

Também conhecido como *OpenStreetMap* o (OSM) é um projeto criado em 2004 com o objetivo de fornecer dados geográficos gratuitos, utilizando um sistema de mapeamento colaborativo, em que as atualizações de rotas e pontos de interesses são feitos por uma comunidade de voluntários os quais são responsáveis por coletar e abastecer a base de dados. Semelhante à ferramenta Google Maps do Google, o OSM possui uma

grande quantidade de informações sobre estradas, trilhos, cafés, estações ferroviárias, etc., de quase todos os locais e regiões do mundo. O site possui uma interface gráfica muito simples, apresentada pela Figura 1.

Figura 1 - Interface Gráfica Openstreetmap



Fonte: openstreetmap.org

Por ser uma ferramenta *Open Source*, o *OpenStreetMap* possibilita o uso indiscriminado de seus mapas sem a necessidade de uma assinatura. Seus dados são publicados usando uma licença aberta *Open Database License*. Atualmente sua base de dados contém o tamanho de 73GB, e está disponível para download em vários formatos, tais como: “.GPX”, “.SHP”, “.OSM”, “.XML” e outros (OpenStreetMap, 2018).

2.3 Problema do Caixeiro Viajante

O Problema do Caixeiro Viajante (PCV) é um clássico problema de otimização combinatória, pertencendo à classe de problemas NP-Difícil (*Non-Deterministic Polynomial Time*), ou seja, sua ordem de complexidade é descrita como não polinomial e comumente são empregados algoritmos, heurísticas e meta-heurísticas para resolução dos mesmos. O PCV se baseia na ideia de que um comerciante tem de percorrer dada quantidade de cidades (n), onde, visando agilizar as entregas, o mesmo terá que percorrer todas as localidades estipuladas e não poderá passar mais de uma vez em determinado

local. Isso tudo no menor espaço de tempo possível. Vale destacar que quanto maior for a rota, mais complexo fica para o algoritmo determinar o melhor trajeto (SILVEIRA, 2000).

Sua estrutura matemática pode ser elaborada através de um grafo em que cada cidade é um nó e as linhas que unem todos os nós são denominadas por arcos. Associada a cada linha está uma distância ou custo.

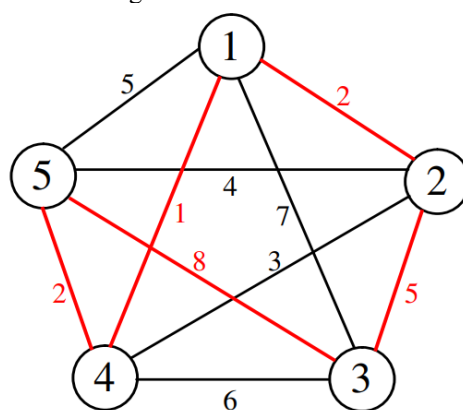
Uma viagem, que passe por todas as cidades uma única vez, corresponde a qualquer subconjunto de linhas do grafo e é designado por circuito Hamiltoniano, na teoria de grafos. O comprimento de um circuito é a soma do comprimento das linhas que fazem parte da viagem (ROCHA, 2006).

Podemos representar o PCV como um conjunto $C = \{c_1, c_2, \dots, c_n\}$ de n cidades c_i e uma matriz de distâncias c_{ij} , cuja distância p é definida por:

- a) $p_{ij} = p(c_i, c_j)$ para $i, j \in \{1, 2, \dots, n\}$;
- b) $p_{ij} = p_{ji}$;
- c) $p_{ii} = 0$.

Neste caso, denominamos o PCV como simétrico, onde as distâncias entre os pontos são as mesmas, independente da direção no qual o viajante caminha, ou seja, $p_{ij} = p_{ji}$ para qualquer par de cidades escolhidas, podendo assim utilizar da lógica Euclidiana para a solução do problema. Na Figura 2 temos sua representação gráfica de um PCV simétrico, onde a solução ideal é dada pela sequência 1-4-5-3-2-1.

Figura 2 - PCV Simétrico

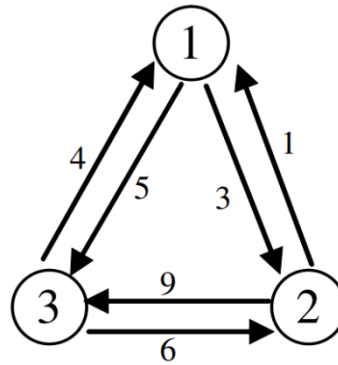


Fonte: ROCHA (2006)

No estudo proposto, as rotas serão elaboradas dentro de uma mesma localidade, contudo, isso não diminui a dificuldade, uma vez que existe uma variação diária dos locais de entrega e uma vasta quantidade de pontos a serem percorridos.

Podemos perceber também que nas cidades, nem todas as ruas possuem mão dupla, ou seja, estamos tratando de um PCV assimétrico, onde para pelo menos um par de pontos de entrega i, j , nós temos $p_{ij} \neq p_{ji}$, não sendo possível tratá-lo com uma lógica Euclidiana simples para a determinação das distâncias entre as casas. A Figura 3 nos mostra graficamente um exemplo de PCV assimétrico.

Figura 3 - PCV Assimétrico



Fonte: ROCHA (2006)

Existem diversas formulações matemáticas para o PCV. Apresentamos aqui a formulação de Dantzig, Fulkerson e Johnson, presente em GOLDBARG e LUNA (2000) visto que a mesma é frequentemente utilizada na literatura, pois apresenta modos peculiares para a caracterização do problema e, também, por ser de fácil compreensão. Sendo $G(N, A)$ o grafo do problema, representado na Figura 3, o modelo pode ser dado por:

$$\text{minimo } z = \sum_{j=1}^n \sum_{i=1}^n p_{ij} x_{ij} \quad (2.1)$$

Sujeito a:

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j \in N \quad (2.2)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \in N \quad (2.3)$$

$$\sum_{ij \in S} x_{ij} \leq |S| - 1 \quad \forall S \in N \quad (2.4)$$

$$x_{ij} \in \{0,1\} \quad \forall i,j \in N \quad (2.5)$$

Onde:

p_{ij} : Custo de ir da cidade i a cidade j (distância);

$x_{ij} = 1$, se arco $(i,j) \in A$, ou seja, for escolhido um caminho válido de i até j ;

$x_{ij} = 0$, se arco $(i,j) \notin A$;

S : subgrafo de G ;

$|S|$: números de vértices do subgrafo S .

Observa-se que essa formulação, x_{ii} não existe, visto que não tem sentido sair de uma cidade e retornar para a mesma. As restrições indicadas em (2.2), determinam que o fluxo de chegada em cada cidade j deve ser 1. As restrições indicadas por (2.3) determinam que o fluxo de saída de cada cidade j deve ser 1. As restrições indicadas por (2.4) impõe a eliminação de circuitos pré-hamiltonianos, ou seja, evitam subciclos. As restrições indicadas por (2.5) determinam que as variáveis sejam binárias, ou seja, podem assumir apenas os valores 0 ou 1.

Devido à natureza do problema, notamos que uma heurística simples não será suficiente para gerar um resultado satisfatório para o problema de otimização de rotas de entrega de encomendas pelos correios. Uma vez que quando se estipular a rota, também é necessário levar em consideração não apenas a distância entre os pontos de entrega, mas também a distribuição das ruas e o sentido das mesmas.

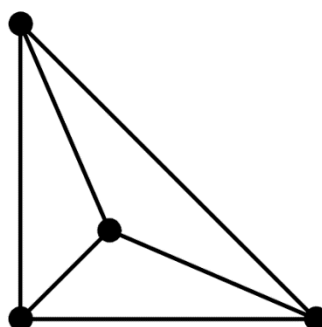
Fora estas diferenças levantadas, outras restrições e adaptações pertinentes ao problema, o algoritmo seguirá a proposta do PCV.

2.4 Algoritmo de Dijkstra

Desenvolvido por Edsger Dijkstra em 1956, o Algoritmo de Dijkstra busca calcular o percurso mais curto em um vértice de grafo dirigido ou não dirigido dentro de um tempo computacional.

Dado um vértice como raiz de busca, o mesmo calcula o custo mínimo deste para os demais vértices do grafo. A ideia deste algoritmo é considerada simples e com bom nível de performance. Contudo, ele não garante uma solução exata em casos de arcos com valor negativo. Para o problema em questão, o mesmo, juntamente com a heurística do caixeiro viajante se tornaram uma excelente alternativa para o problema levantado. Onde, para encontrar a melhor rota, também é necessário levar em consideração os fatores já citados (MÉNDEZ; GUARDIA, 2008). Na Figura 4 temos a representação de um grafo, sendo este composto por: quatro vértices e seis arestas.

Figura 4 - Representação de um grafo

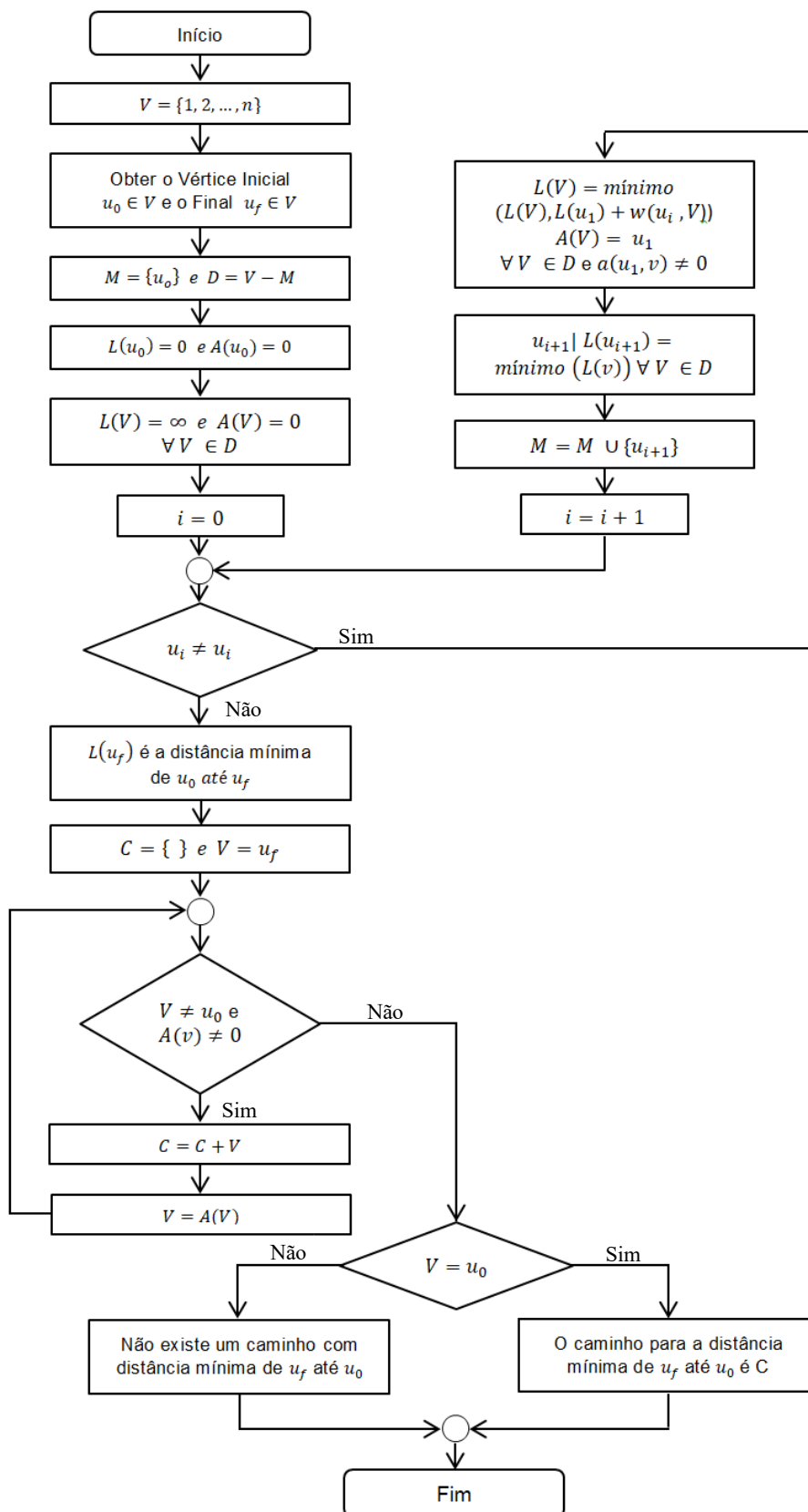


Fonte: O autor

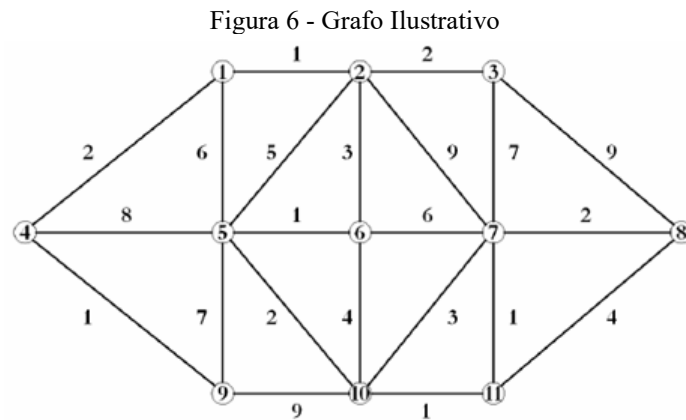
Inicialmente os vértices do grafo em estudo devem ser numerados, possibilitando a elaboração de uma matriz de dados que pode ser usada como base para o desenvolvimento da lógica do algoritmo. O passo seguinte é definir respectivamente o vértice origem (u_0) e o vértice destino (u_f) nesta matriz de dados que denominamos Matriz de Vínculos e Pesos.

De modo geral, qualquer situação que possa ser representada por um grafo simples e que possua uma matriz de vínculos e pesos definida é passível de ser submetido à proposta de Dijkstra. Na Figura 5 temos um fluxograma que nos mostra o seu princípio de funcionamento.

Figura 5 - Fluxograma do Algoritmo de Dijkstra



Na Figura 6 temos a apresentação de um grafo ilustrativo cuja origem é o vértice 4 ($u_0 = 4$) e o destino é o vértice 8 ($u_f = 8$) que utilizaremos como exemplo para detalhar o funcionamento do algoritmo.



Fonte: Barros; Pamboukian e Zamboni (2007)

A matriz de vínculos e pesos pode ser composta por informações que representam distância entre localidades, custos operacionais, quantidade de recursos, ou qualquer outro elemento passível de ser ponderado entre os vértices (ZAMBONI; MONEZZI, 2002). A Figura 7 nos mostra a representação da Matriz de Vínculos e Pesos do Grafo Ilustrativo.

Figura 7 - Matriz de Vínculos e Pesos

Matriz de Vínculos e Pesos											
$\begin{smallmatrix} L \\ C \end{smallmatrix}$	1	2	3	4	5	6	7	8	9	10	11
1	0	1	0	2	6	0	0	0	0	0	0
2	1	0	2	0	5	3	9	0	0	0	0
3	0	2	0	0	0	0	7	9	0	0	0
4	2	0	0	0	8	0	0	0	1	0	0
5	6	5	0	8	0	1	0	0	7	2	0
6	0	3	0	0	1	0	6	0	0	4	0
7	0	9	7	0	0	6	0	2	0	3	1
8	0	0	9	0	0	0	2	0	0	0	4
9	0	0	0	1	7	0	0	0	0	9	0
10	0	0	0	0	2	4	3	0	9	0	1
11	0	0	0	0	0	0	1	4	0	1	0

Fonte: Barros; Pamboukian e Zamboni (2007)

Com base no grafo e nos dados da Matriz de Vínculos e Pesos, criamos dois conjuntos de vértices, denominados Vértices Marcados (M) e Vértices Desmarcados (D).

A marcação destes vértices por ser feita por meio de um vetor com dados do tipo booleano (*bool*), verdadeiro (*true*) ou falso (*false*), dessa forma os vértices marcados podem assumir o valor verdadeiro e os vértices desmarcados assumem o valor falso.

Na sequência, são criados dois vetores para definir os comprimentos acumulados (L) e o vértice anterior (A), sendo que ambos guardam informações relativas ao nó em estudo. O vetor Comprimento (L) deve ser iniciado com valores máximos ou outra forma que possa sinalizar que o mesmo possui valores infinitos, enquanto que o vetor Vértice Anterior (A) é iniciado com valores nulos.

Após este procedimento inicial de preparação das estruturas de dados, é dado início a uma rotina de verificação vértice a vértice, partindo sempre do vértice inicial (u_0) até atingir o vértice de destino (u_f). Esta rotina de interação consiste basicamente dos seguintes passos:

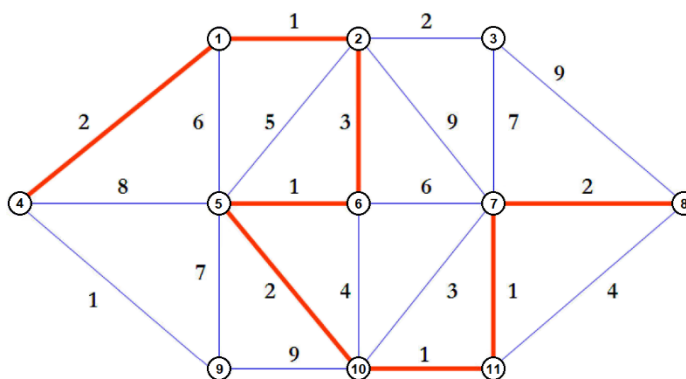
- a) Verificar as conexões do vértice atual (u_i) que não foram usadas (desmarcadas);
- b) Calcular as distâncias acumuladas das conexões disponíveis;
- c) Encontrar entre as conexões disponíveis qual apresenta a menor distância acumulada e marcá-la;
- d) Repetir a rotina até o vértice atual ser o vértice de destino ($u_i = u_f$).

Uma vez definidas as distâncias mínimas, executa-se uma rotina para varrer os vértices anteriores partindo-se do vértice destino (u_f), que é a sequência em ordem inversa dos caminhos mínimos.

Finalizando, o algoritmo ainda contempla uma verificação que atua caso não exista uma conexão válida entre os vértices inicial (u_0) e final (u_f).

Na Figura 8, tem-se novamente a apresentação do Grafo Ilustrativo, já com o caminho traçado pelo algoritmo de Dijkstra, calculado com base na sua respectiva Matriz de Vínculos e Pesos.

Figura 8 - Grafo Ilustrativo com o caminho mínimo calculado



Fonte: Barros; Pamboukian e Zamboni (2007)

Assim, verifica-se que qualquer situação que possa ser representada por um grafo simples e que possua uma matriz de vínculos e pesos bem definida, são passíveis de serem submetidas à proposta de solução do Algoritmo de Dijkstra.

2.5 Algoritmos Genéticos Híbridos

Os Algoritmos Genéticos (AGs) são métodos de busca e otimização baseados na teoria evolutiva de Charles Darwin. Os AGs tentam imitar os passos da seleção natural de maneira a gerar populações de indivíduos mais adaptados as limitações e dificuldades do ambiente em que habitam ou ao problema ao qual pretendem solucionar.

Os mesmos são bastante empregados na busca de soluções ou na minimização de problemas considerados demasiadamente complexos. Como exemplo de utilização, pode ser citado o seu emprego na busca da menor distância entre uma sequência de pontos (CORMEN et al., 2002). A seguir temos os passos típicos de um algoritmo genético, sendo eles:

- Gera-se a população inicial;
- Avalia-se cada indivíduo de acordo com a função objetivo;
- Utiliza-se um método de seleção para reproduzir os indivíduos;
- Utiliza-se um método de cruzamento para gerar uma nova população, de acordo com a taxa de cruzamento;
- Aplica-se o operador de mutação aos indivíduos da nova população que foram escolhidos, de acordo com a taxa de mutação;

- f) Finaliza a execução do algoritmo de acordo com um critério de parada, ou retorna-se ao passo 2 (CORMEN et al., 2002).

Existem vários critérios de parada e entre os mais utilizados estão: a definição de um número fixo de gerações ou executar o algoritmo até que o indivíduo com menor pontuação atinja um resultado pré-estabelecido. Há várias técnicas que podem ser empregadas para simular os passos do algoritmo e nas próximas subseções serão comentadas as que foram adotadas, justificando sua abordagem no presente trabalho.

A definição de Algoritmo Genético Híbrido neste estudo vem do fato de se utilizar o Algoritmo de Dijkstra como critério de avaliação para cada indivíduo.

2.5.1 Representação do Cromossomo

A representação cromossômica é essencial para o algoritmo genético e consiste em uma maneira de interpretar as informações do problema em uma expressão funcional que possa ser tratada pelo computador. A adequação da representação ao problema implica diretamente na qualidade dos resultados gerados pelo algoritmo. Uma vez escolhida a representação cromossômica, essa escolha implica diretamente no uso dos operadores genéticos (RODRIGUES, 2004).

A definição da representação cromossômica fica a critério do programador, desde que esta esteja adequada ao problema. Entretanto, existem algumas regras gerais que devem ser levadas em consideração, são elas:

- a) A representação deve ser a mais simples possível;
- b) Se houver soluções proibidas ao problema, então elas não devem ter uma representação;
- c) Se o problema impuser condições de algum tipo, estas devem estar implícitas dentro da representação (LINDEN, 2008, p. 59).

Seguindo estes conceitos a representação cromossômica simboliza as possíveis soluções do problema, geralmente representados por vetores de soluções, que de acordo

com o problema assumem diferentes estruturas, dentre elas, existe a binária, a qual é descrita como a mais simples e utilizada na literatura dos algoritmos genéticos.

A representação binária nada mais é do que um cromossomo (vetor) formado por uma sequência de bits (cada bit um gene), onde os valores destes são compostos pelos valores 1 ou 0 (RODRIGUES, 2004). Outra forma de representação, sendo esta mais utilizada em problemas de otimização combinatória, é a representação cromossômica por lista.

2.5.2 Operadores Genéticos

Os operadores genéticos consistem em aproximações computacionais de fenômenos vistos na natureza, como: a reprodução sexuada, a mutação genética e quaisquer outros que a imaginação dos programadores consiga reproduzir (LINDEN, 2008, p. 41). Os mesmos são responsáveis por alterar as populações durante sucessivas gerações para que o AG consiga encontrar valores satisfatórios para a resolução do problema. Borges (2003) define os seguintes operadores genéticos:

- a) Operador de seleção: faz a seleção dos pais mais aptos a fim de reproduzir membros da população que tenham condições de atingir a função objetivo;
- b) Operador de mutação: são necessários para a introdução e manutenção da diversidade genética da população, alterando um ou mais componentes de uma estrutura escolhida. Desta forma, a mutação assegura que a probabilidade de se chegar a qualquer ponto do espaço de solução nunca seja zero;
- c) Operador de cruzamento: é responsável pela recombinação das características dos pais durante a reprodução, permitindo assim a herança das características dos pais.

Existem vários operadores de seleção, entre os mais utilizados estão os baseados em probabilidade, ranking e torneio. Dentre os baseados em probabilidade destaca-se o método da Seleção Proporcional. Na seleção proporcional ou Roleta, cada indivíduo “i” possui uma probabilidade de sobrevivência de acordo com a sua aptidão, ou seja, quanto maior a aptidão do indivíduo, maior será a chance dele de ser escolhido para participar do cruzamento (LINDEN, 2008).

O operador de cruzamento em problemas de otimização combinatória não é o mesmo empregado na representação binária, pois este processo poderia resultar na geração de filhos inválidos, ou seja, indivíduos que violariam as restrições impostas pelo problema. Para evitar que filhos inválidos sejam gerados é necessário a adoção de outros tipos de heurística de cruzamento.

A mutação do cromossomo tem associada uma probabilidade extremamente baixa pois, se ela tiver um valor muito alto, o algoritmo genético se parecerá muito com uma técnica chamada *random walk*, na qual a solução é determinada de forma aleatória (simplesmente sorteando-se elementos sem usar informações correntes ou passadas) (LINDEN, 2008).

Depois disso é calculada a avaliação da população para saber quem é o melhor indivíduo. Este processo se repete até a quantidade de gerações definidas pelo usuário (VIEIRA, 2013).

À medida que as rotas são traçadas pelo AG, alguns problemas podem ocorrer. Dentre eles, destaca-se que as rotas geradas podem eventualmente se cruzar. Para solucionar este problema, existem algumas heurísticas de melhoramento que podem ser utilizadas, uma delas é o 2-OPT. Proposto por LINS em 1973, o 2-OPT é uma heurística que busca, através de uma solução inicial, fazer trocas entre as rotas, a fim de diminuir seu custo original, ou seja, quando um cruzamento de rotas é identificado o 2-OPT é usado para desfazê-los. Por meio da troca das ligações entre as casas, a heurística cria novas rotas com percursos alternativos. Se o custo da nova rota for menor que o da original, a rota é mantida. Este método é executado até que permaneça a menor rota possível para as casas selecionadas (GOLDBARG; LUNA, 2000).

2.5.3 Avaliação

A aptidão do indivíduo é medida de acordo com a sua função de avaliação. A função de avaliação é a maneira utilizada pelos Algoritmos Genéticos para estipular as chances de um indivíduo ser a solução do problema a ser tratado (LINDEN, 2008). Pode-se entendê-la mais facilmente ao se olhar para a função de avaliação como sendo a nota dada a um indivíduo na resolução de um problema. Esta nota será usada para a escolha dos indivíduos, sendo a forma de diferenciar entre as boas e as más soluções para o problema,

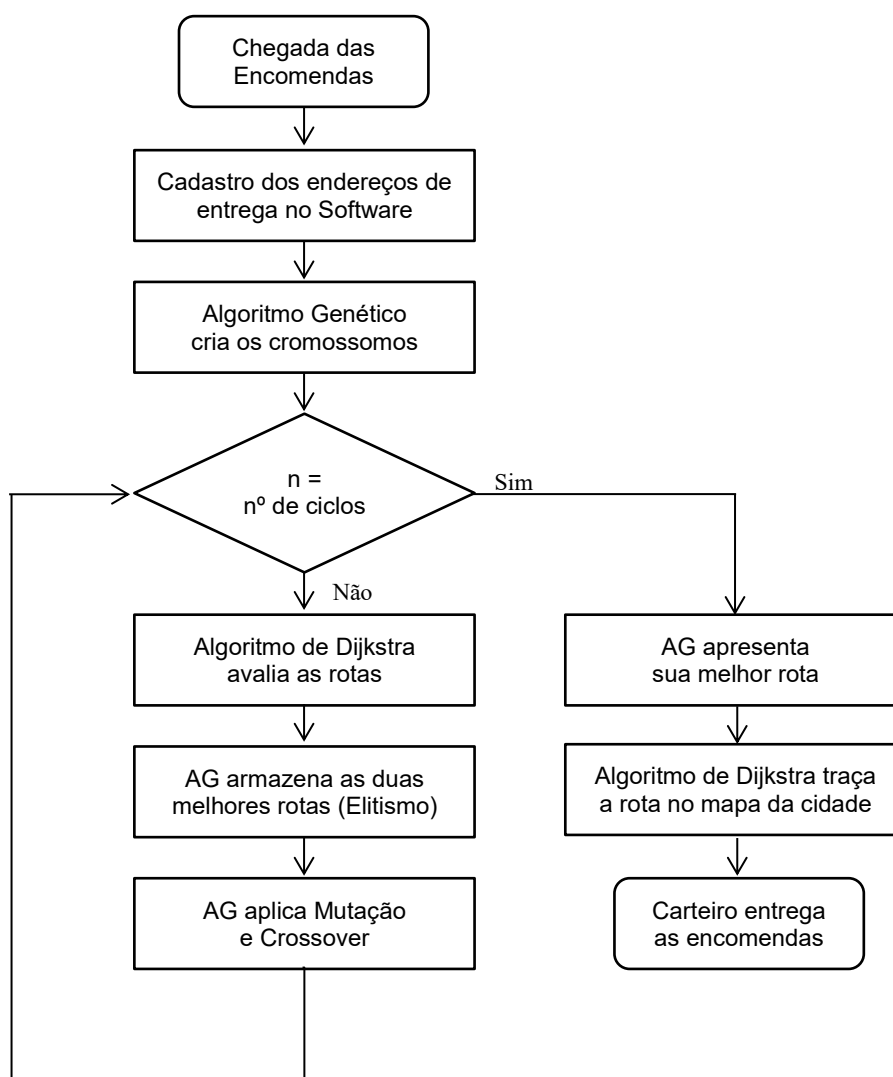
ou seja, se o cromossomo C_1 representar uma solução melhor do que o cromossomo C_2 , então a avaliação do cromossomo C_1 deverá ser maior do que a avaliação de C_2

A função de avaliação, em muitos casos, é a única ligação entre o programa e o problema real. Isto ocorre porque a função de avaliação só julga a qualidade da solução que está sendo apresentada por aquele indivíduo, não armazenando qualquer tipo de informação sobre as técnicas utilizadas para a resolução do problema.

3 METODOLOGIA

Este capítulo apresenta de forma detalhada as técnicas aplicadas no desenvolvimento do projeto, visando solucionar o problema em questão. A figura 9 apresenta um fluxograma básico do funcionamento do sistema de entregas das encomendas proposto por este estudo.

Figura 9 - Fluxograma básico do sistema de entregas proposto



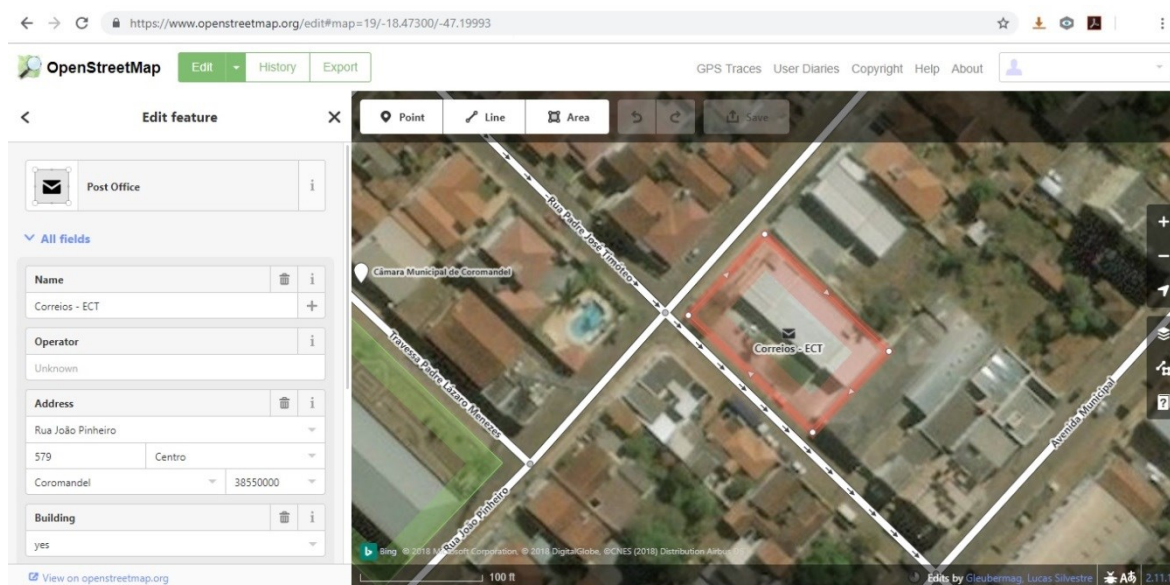
Fonte: O autor

3.1 OpenStreetMap

Possuir uma interface gráfica amigável ao usuário é parte fundamental para o desenvolvimento do projeto, visto que nela deve conter o mapa da cidade com ruas, avenidas e demais informações que possam ajudar o usuário final a identificar facilmente sua rota e os pontos de entrega das encomendas.

Semelhante a outros sites de navegação disponíveis na internet, o OSM possui uma interface de uso muito simples, apresentada na Figura 10. Nela pode-se observar a localização exata da Empresa de Correios e Telégrafos da cidade de Coromandel-MG, situada à Rua João Pinheiro, 579.

Figura 10 - Vista por satélite da ECT de Coromandel-MG



Fonte: openstreetmap.org (2018)

A interface apresentada disponibiliza ao usuário a opção de exportar as coordenadas geográficas e os dados de navegação de qualquer parte do mundo. No estudo de caso proposto, exportamos os dados referentes ao município de Coromandel - MG.

Exportar os dados possibilita navegar para qualquer ponto da cidade sem a necessidade de conexão com a internet durante o trabalho de entrega das encomendas, diminuindo consideravelmente os custos de desenvolvimento e implantação do projeto.

As informações de Latitude e Longitude contidas na base de dados foi o ponto de partida para o desenvolvimento do Algoritmo Genético desenvolvido para a solução do problema.

Estas informações foram extraídas da base de dados do site utilizando a biblioteca *Open Street Map Functions for MATLAB*, que é um conjunto de funções desenvolvidas por Ioannis F. Filippidis, da Universidade Técnica Nacional de Atenas, Grécia, para importar informações do arquivo de dados XML do *OpenStreetMap*.

O formato XML é carregado primeiro no MATLAB® e, em seguida, um script de análise extrai parte das informações contidas no arquivo. Isso é salvo em uma estrutura do MATLAB® e a partir deste ponto, os dados podem ser usados para traçar a rede de transporte e elaboração das rotas.

Na Figura 11 podemos ver uma parte da tabela esparsa, extraída do banco de dados do *OpenStreetMap*, que representa os pontos no mapa e suas respectivas interligações. Observe que o ponto 374 neste caso se trata da localização dos correios.

Figura 11 - Parte da matriz esparsa que representa os pontos de interligação das ruas

```

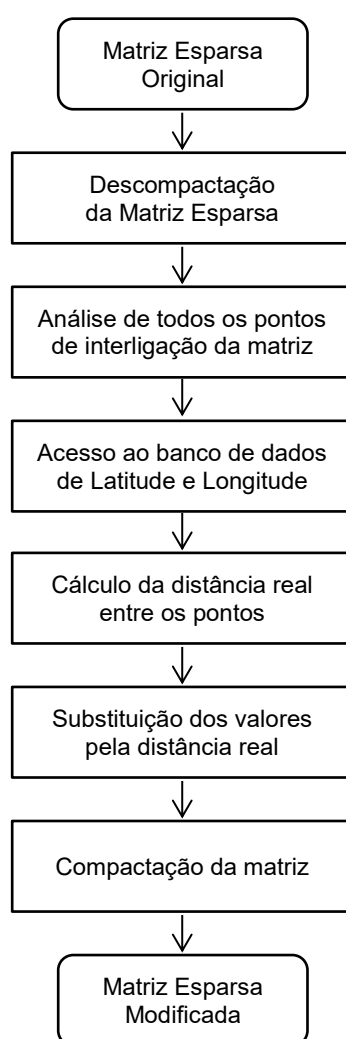
...
...
(745,372)  1
(283,373)  1
(291,373)  1
(304,374)  1
(802,374)  1
(803,374)  1
(941,374)  1
(945,374)  1
(374,375)  1
(802,375)  1
(803,375)  1
...
...

```

Fonte: openstreetmap.org (2018)

Observando a matriz esparsa apresentada, nota-se que os valores entre as conexões dos pontos são todos unitários, representando apenas a existência da conexão. Por isso, foi necessário um pré-processamento neste banco de dados antes de iniciar a sua utilização, de modo a calcular a distância real entre os pontos de conexão e utilizar estas distâncias na determinação da melhor rota pelo algoritmo de Dijkstra. A Figura 12 nos mostra, em forma de fluxograma, este pré-processamento.

Figura 12 - Fluxograma do pré-processamento do banco de dados

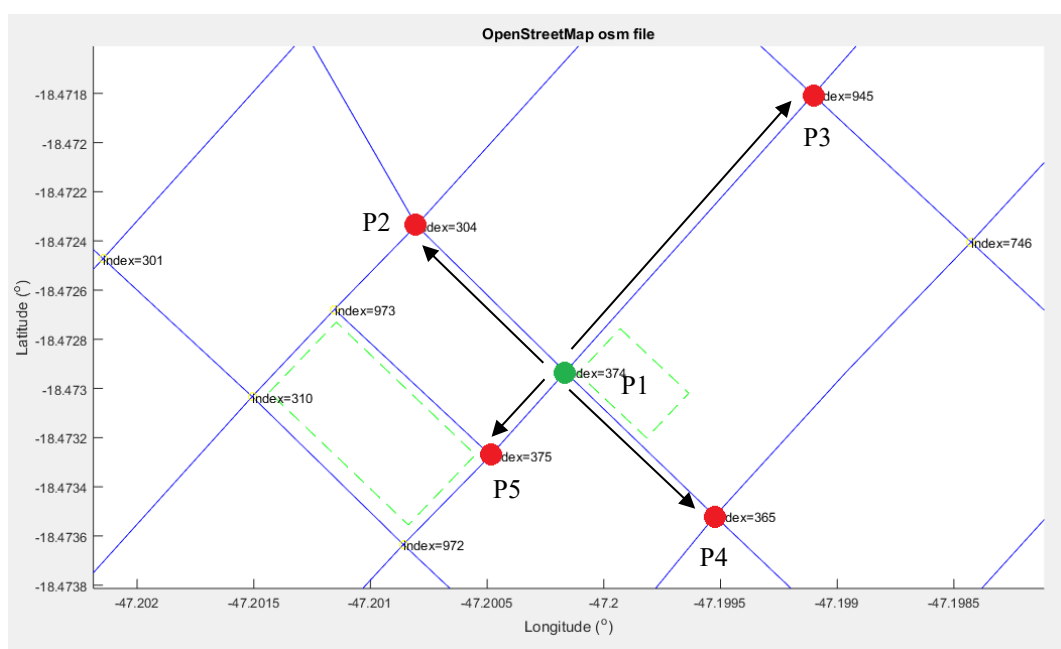


Fonte: O autor

Analisando o ponto P1, que representa graficamente a localização da Empresa de Correios e Telégrafos, cujo índice na matriz de vínculos e pesos é igual a 374, podemos visualizar através da Figura 13, seus respectivos pontos de interligação possíveis de serem utilizados para a elaboração das rotas, que são eles:

- Ponto P2 (índice 304)
- Ponto P3 (índice 945)
- Ponto P4 (índice 365)
- Ponto P5 (índice 375)

Figura 13 - Pontos de Interligação da Agência dos Correios



Fonte: O Autor

Pode-se perceber que a distância real entre os Pontos P1 (índice 374) e P3 (índice 945) é a maior entre todas as conexões disponíveis, totalizando uma distância de 154,847 metros. Da mesma forma, podemos afirmar que a distância real entre os pontos P1 (índice 374) e o Ponto P5 é a menor entre todas, totalizando 46,625 metros.

As distâncias entre os pontos P1 (índice 374) e P2 (índice 304) e entre os pontos P1 (índice 374) e P4 (índice 365) apresentam visualmente valores próximos, o que ficou comprovado com a resposta do pré-processamento que nos forneceu 87,335 metros entre P1 e P2 e 86,849 metros entre P1 e P4.

A Figura 14, nos mostra os dados da matriz esparsa, modificada pelo pré-processamento, contendo as distâncias calculadas em metros, dos pontos aos quais existem conexão com o índice 374, que representa a localização dos correios.

Figura 14 - Matriz Esparsa Modificada: Conexões do Ponto 374

...	
...	
(304,374)	87,335
(945,374)	154,847
(365,374)	86,849
(375,374)	46,625
...	
...	

Fonte: O Autor

Percebe-se agora que a matriz de vínculos e pesos deixou de ser binária, e que nos apresenta agora a informação da distância real em metros entre os pontos aos quais existem uma conexão válida.

3.2 Desenvolvimento do Algoritmo Genético Híbrido

Nesta subseção, apresentamos de forma detalhada as técnicas e heurísticas aplicadas no desenvolvimento do algoritmo genético híbrido utilizado para solucionar o problema proposto.

3.2.1 Arquitetura do Cromossomo

Conforme já mencionado, a adequação da representação do cromossomo ao problema implica diretamente na qualidade dos resultados gerados pelo algoritmo criado.

Para a otimização do percurso de distribuição de encomendas dos correios, a solução está relacionada com a ordem que os elementos estão dispostos.

Cada elemento, ou encomenda, é associado a uma casa a ser visitada pelo carteiro, ao qual corresponde a um gene do cromossomo $C_{i,j}$. As casas são dispostas na ordem em que serão visitadas, como por exemplo: partindo do depósito $C_{0,0}$, o carteiro deve visitar o ponto $C_{0,1}$, $C_{1,3}$, $C_{3,4}$, $C_{4,6}$, $C_{5,2}$, $C_{2,1}$ e, uma vez entregue todas as suas encomendas, o carteiro deve retornar ao ponto de partida. Na Figura 15 temos a representação de um cromossomo válido.

Figura 15 - Representação de um cromossomo válido

1	3	4	5	2	1
---	---	---	---	---	---

Fonte: O autor.

O carteiro retornar ao seu ponto de partida, sede dos correios, é um fator importantíssimo para a elaboração da rota, uma vez que, independente do número de encomendas a serem entregues, o ponto de partida e o ponto de chegada deverão ser sempre os mesmos. Portanto, o cromossomo só poderá ser considerado válido, se iniciar e terminar no ponto que denominaremos de P_1 .

3.2.2 Operador de Seleção

O operador de seleção do algoritmo genético elaborado foi baseado no princípio do elitismo.

O elitismo é uma pequena modificação no módulo da população que quase não altera o tempo de processamento do algoritmo, mas que garante que o desempenho do AG sempre cresça com o decorrer das gerações.

A ideia básica por trás do elitismo é a seguinte: os n melhores indivíduos de cada geração não devem “morrer” junto com a sua geração, mas sim passar para a próxima visando garantir que seus genomas sejam preservados (LINDEN, 2008, p. 141).

A cada ciclo de processamento do AG, o cromossomo que apresentar o melhor resultado, ou seja, o caminho que apresentar a menor distância é preservado e tem a sua

existência garantida no próximo ciclo do processamento. Uma vez que o AG encontre outro caminho que apresente uma distância ainda menor do que a já encontrada, este é preservado e encaminhado para o próximo ciclo, até que a quantidade de ciclos ou outro parâmetro de parada de processamento do AG seja atingido.

3.2.3 Operador de Cruzamento

O operador de cruzamento escolhido para o desenvolvimento do algoritmo genético foi um cruzamento parcial denominado *Partially Mapped Crossover* (PMX).

Proposto por Goldberg e Lingle em 1985, este método utiliza dois pontos de corte. Para cada par de pais, são escolhidos dois pontos de corte aleatórios e permutado o material genético contido entre eles para gerar os filhos, caso algum filho seja inválido utiliza-se uma matriz de mapeamento que representa os valores dos pais que serão trocados caso haja valores repetidos no cromossomo (VIEIRA, 2013). A Figura 16 representa graficamente o funcionamento deste operador.

Figura 16 - Representação gráfica do cruzamento PMX



Fonte: O autor.

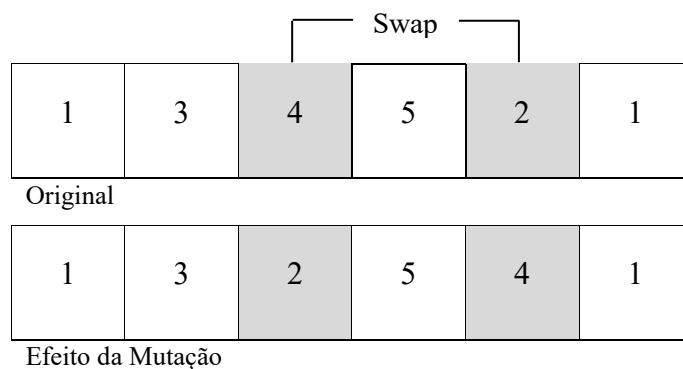
Para preservar a validade do cromossomo, adiciona-se uma restrição ao operador de cruzamento, de modo que o primeiro ponto de corte não escolha o primeiro ponto do

cromossomo e o segundo ponto de corte nunca seja no último ponto. Desta forma, evita-se que o cruzamento gere resultados inválidos.

3.2.4 Operador de Mutação

O operador de mutação escolhido para o algoritmo genético foi o *SWAP Mutation*. Este operador, como o próprio nome o define, efetua uma permuta ou troca. Nele são sorteados dois valores aleatórios que correspondem à posição do vetor que representa o cromossomo. Estes dois valores definirão onde os pontos de entrega do carteiro serão trocados de posição. A Figura 17 representa graficamente o funcionamento do operador de mutação do tipo *Swap*.

Figura 17 - Representação gráfica da Mutação Swap



Fonte: O autor.

No desenvolvimento do operador de mutação, foi necessário adicionar uma restrição ao seu funcionamento. Uma vez que o carteiro deve sair da sede dos correios e retornar ao mesmo lugar ao final das entregas, ficou proibida a troca ou permuta do ponto de partida e do ponto de chegada. Desta forma, a mutação será aplicada somente aos pontos de entrega selecionados.

3.2.5 Avaliação

A função de avaliação geralmente utilizada para o problema em questão é calculada através da fórmula do acumulado das distâncias Euclidianas entre todos os pontos do trajeto.

$$\sum_{i=0}^{n-1} \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3.1)$$

Onde:

- x_1 - Coordenada x do ponto P_1 .
- x_2 - Coordenada x do ponto P_2 .
- y_1 - Coordenada y do ponto P_1 .
- y_2 - Coordenada y do ponto P_2 .
- n - Número total de Pontos.
- i - Contador do número de pontos.

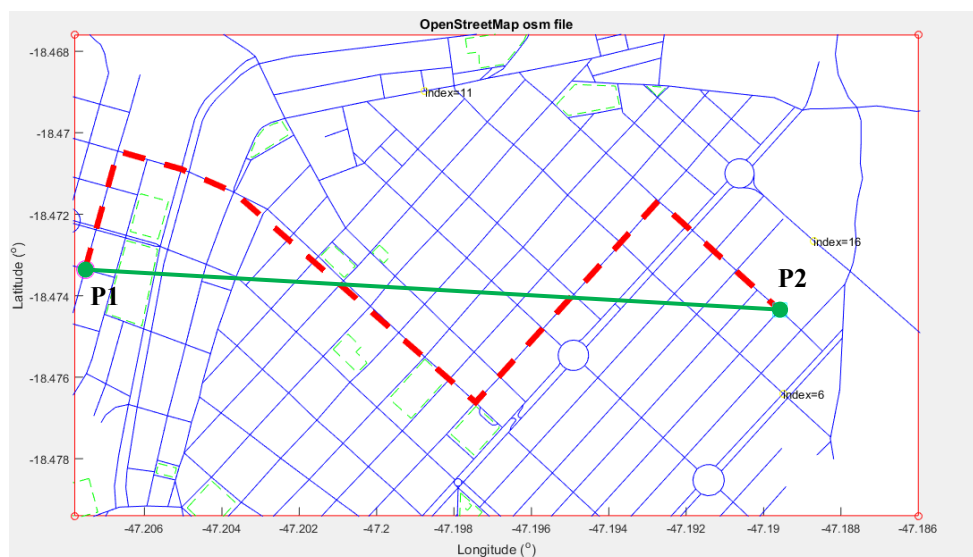
As distâncias entre as casas são medidas através de coordenadas geográficas (latitude e longitude), aos quais são extraídas de um arquivo XML importado da base de dados do OpenStreetMap.

Com base nestas coordenadas, podemos fazer uma aproximação e projetar a localização destas casas e das ruas em um plano cartesiano. Desta forma podemos determinar as distâncias Euclidianas ponto a ponto seguindo a sequência de cada rota e determinar a pontuação de cada cromossomo.

Podemos perceber que este tipo de avaliação, não considera as características das ruas e avenidas, podendo gerar um resultado insatisfatório no momento da apresentação da rota.

A Figura 18 apresenta na linha contínua, a distância Euclidiana entre os Pontos P_1 e P_2 e na linha tracejada, a rota obedecendo às características reais das ruas e avenidas, traçadas pelo algoritmo de Dijkstra.

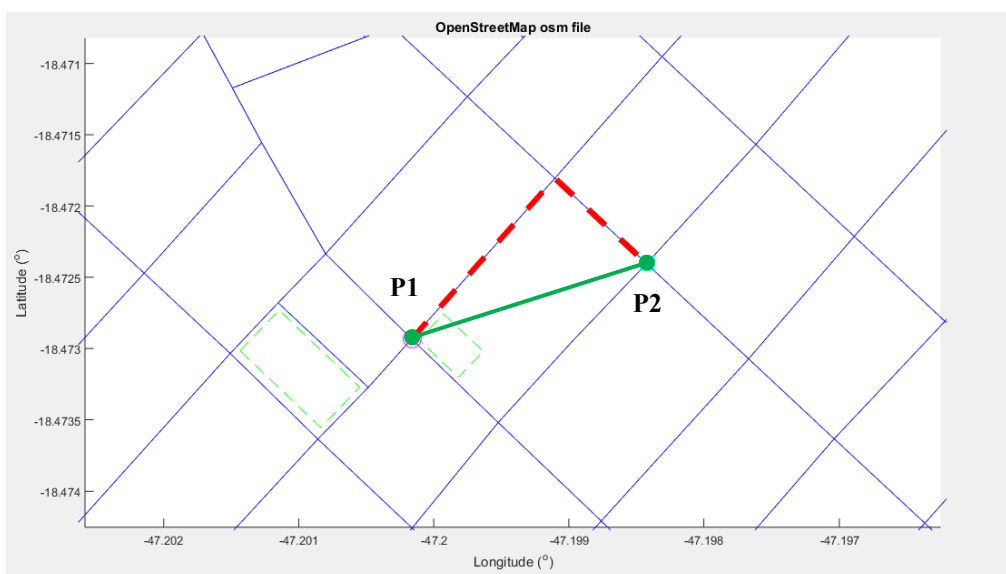
Figura 18 - Avaliação das distâncias entre P1 e P2



Fonte: O autor

A Figura 19 apresenta também na linha contínua, a distância Euclidiana entre os pontos P1 e P2 e na linha tracejada, a rota obedecendo às características reais das ruas e avenidas, traçadas pelo algoritmo de Dijkstra. Percebe-se que até para as rotas aparentemente simples, se o critério de avaliação for feito apenas com a distância Euclidiana, os resultados não serão satisfatórios.

Figura 19 - Distâncias entre P1 e P2



Fonte: O Autor

Com base nestes resultados, percebe-se que a utilização do critério de avaliação utilizando a distância Euclidiana não seria uma proposta muito interessante para a resolução do problema.

Por outro lado, utilizando a capacidade do Algoritmo de Dijkstra como opção de avaliação do Algoritmo Genético, cria-se um meio que possibilita ao AG estabelecer critérios e levar em consideração as características e obstáculos reais das ruas e avenidas no momento de avaliar as rotas de cada cromossomo.

Um ponto importante a ser destacado neste tipo de avaliação é que as distâncias não são mais calculadas ponto a ponto no plano cartesiano.

Sempre respeitando a sequência sugerida pelo AG para cada indivíduo, o Algoritmo de Dijkstra atua traçando e calculando as distâncias das rotas entre todas as casas a serem visitadas, e a somatória das distâncias reais de cada rota utilizadas como pontuação de cada indivíduo do AG.

A função de avaliação continua sendo feita através da fórmula do acumulado das distâncias entre todos os pontos do trajeto, porém esta distância agora é dada diretamente pelo Algoritmo de Dijkstra, restando ao AG executar somente a sua somatória.

$$\sum_{i=0}^{n-1} d_i \quad (3.2)$$

Onde:

d_i - Distância entre as casas da rota.

n - Número de casas da rota.

i - Contador do número de pontos.

Assim, somando as vantagens do Algoritmo Genético e do Algoritmo de Dijkstra, criamos um Algoritmo Genético Híbrido.

4 TRABALHOS RELACIONADOS

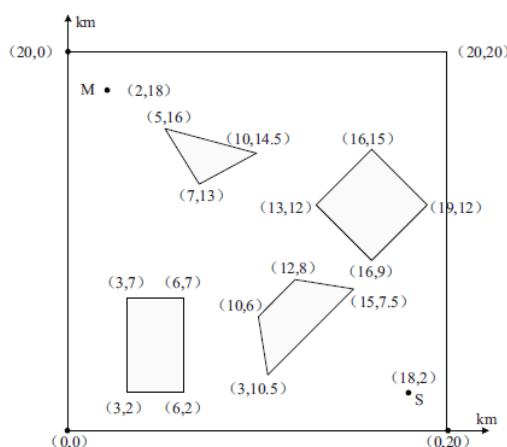
Neste capítulo são apresentados alguns trabalhos dos quais seus autores utilizaram-se de técnicas semelhantes às principais técnicas envolvidas na elaboração do projeto proposto.

4.1 Algoritmos Híbridos

Sabe-se que muitos problemas complexos podem ser decompostos em problemas menores e resolvidos com técnicas distintas. Trabalhos como os de (JINZHU; JIANGUO; QUN, 2014), que utiliza a combinação do Diagrama de Voronoi, Algoritmo Genético e Algoritmo de Dijkstra, para calcular a rota de um míssil do tipo navio-navio nos mostra a versatilidade que a combinação de algoritmos pode nos oferecer.

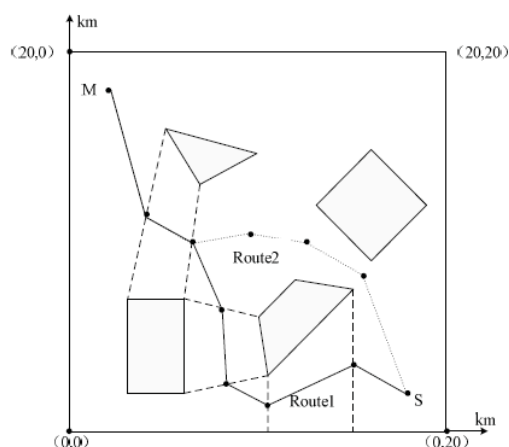
A Figura 20 apresenta o diagrama de Voronoi criado a partir das coordenadas dos obstáculos entre os pontos M e S, enquanto que a Figura 21 apresenta duas opções de rotas para o míssil, calculadas com o auxílio do algoritmo de Dijkstra.

Figura 20 – Diagrama de Voronoi



Fonte: Jinzhu; Jianguo e Qun (2014)

Figura 21 – Esboço do Mapa da Rota



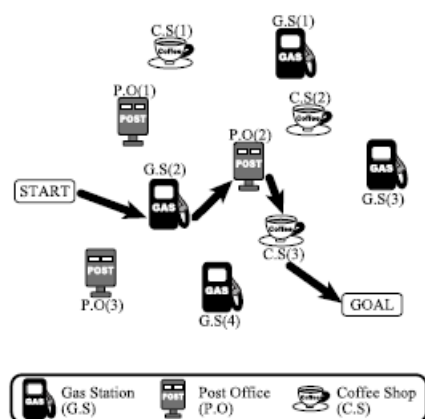
Fonte: Jinzhu; Jianguo e Qun (2014)

Devido ao fato do problema proposto nesta dissertação se tratar de um problema semelhante ao Caixeiro Viajante Assimétrico, houve também a necessidade de utilizar técnicas distintas para viabilizar a solução do problema, sendo o Algoritmo Genético e o Algoritmo de Dijkstra as duas principais técnicas escolhidas.

4.2 Algoritmo de Dijkstra como critério de avaliação

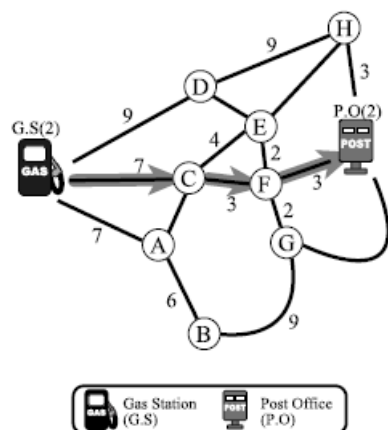
O trabalho de (YOSHIKAWA; TERAÍ, 2009), apresenta um sistema de navegação para carros que permite encontrar um percurso com vários pontos de passagem antes de chegar ao seu destino. Na Figura 22 temos a representação de um posto de gasolina, uma agência de correios e uma cafeteria, que deverão ser visitadas antes do motorista chegar ao seu destino, e na Figura 23 temos uma representação mais detalhada da rota entre o posto de gasolina e a agência dos correios.

Figura 22 – Seleção do Ponto de Passagem



Fonte: Yoshikawa e Terai (2009)

Figura 23 – Detalhe do Roteamento



Fonte: Yoshikawa e Terai (2009)

Neste problema, obrigatoriamente o motorista deve passar por um posto de combustível, por uma agência dos correios e em uma cafeteria, não necessariamente nesta ordem, antes de chegar ao seu destino. A seleção de qual posto de gasolina, qual agência dos correios e qual cafeteria a ser escolhida ficou a critério do Algoritmo Genético, enquanto que a distância entre eles foi determinada com o uso do algoritmo de Dijkstra.

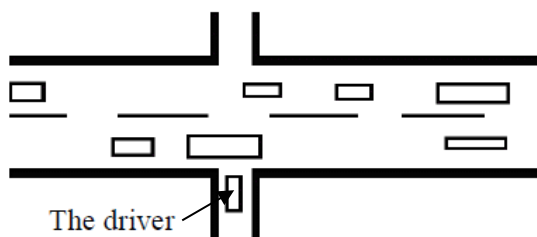
De forma bem semelhante ao proposto por este trabalho, temos que o AG tenta encontrar soluções viáveis para o problema, enquanto que o algoritmo de Dijkstra avalia as rotas e informa sobre as características de cada uma delas.

4.3 Ajustes na Matriz de Vínculos e Pesos

Em outro problema de roteamento para veículos descrito em (KAMBAYASHI, Y. et. al, 2009), os autores resolveram ajustar os valores da matriz de vínculos e pesos utilizada pelo algoritmo de Dijkstra, de modo a solucionar o problema não somente em busca de um caminho mais curto, mas sim em busca também de uma rota mais confortável para o motorista.

Além da distância total a ser percorrida no trajeto, eles consideraram por exemplo, que uma rota que opte por cruzar por uma avenida movimentada e que não possua semáforo para a sua travessia, deva possuir um peso maior do que uma rota que evite este tipo de caminho. A Figura 24 nos mostra um exemplo deste tipo de manobra, no qual o motorista deverá aguardar um momento em que as duas vias principais estejam sem trânsito para então conseguir atravessar até o outro lado.

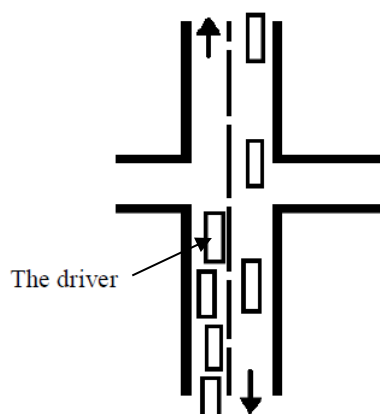
Figura 24 – Cruzamento sem sinal



Fonte: Kambayashi et al. (2009)

Lembrando que no Japão os veículos circulam pelo lado esquerdo das ruas, os autores consideraram também que executar uma conversão para o lado direito é mais vantajoso do que fazer uma conversão para o lado esquerdo, visto que para executar esta segunda manobra, o motorista deverá aguardar a liberação da pista da esquerda para realizá-la. Na Figura 25 temos um exemplo deste tipo de manobra, onde o motorista se encontra em uma via de mão dupla e têm a possibilidade de executar uma conversão tanto para direita, mais vantajoso, quanto para a esquerda, menos vantajoso.

Figura 25 – Conversão à direita ou à Esquerda



Fonte: Kambayashi et al. (2009)

Além destas duas penalidades apresentadas, foram criadas várias outras, sempre incrementando valores na matriz de vínculos e pesos, com o objetivo de penalizar as manobras definidas como menos confortáveis para o condutor do veículo, conforme critério estabelecido pelos autores do projeto.

Neste exemplo, temos uma demonstração de que a ideia de modificar ou ajustar os dados da matriz de vínculos e pesos utilizada pelo algoritmo de Dijkstra pode nos proporcionar resultados ainda melhores do que os resultados obtidos apenas com o uso dos dados originais da matriz. Deve-se considerar que o contrário também é verdadeiro, e que modificações que não respeitem critérios bem estabelecidos podem apresentar resultados nada animadores.

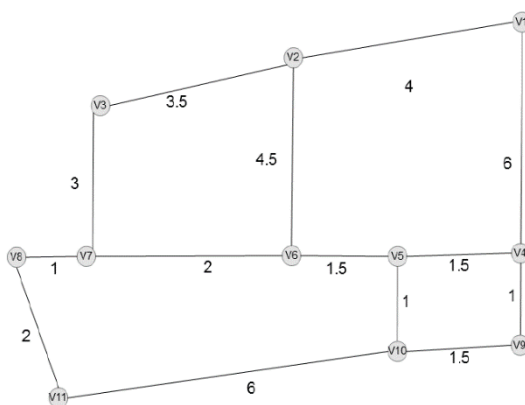
Na busca da solução para um roteamento que leve em consideração o veículo a ser utilizado para a entrega de encomendas proposto por este trabalho, a modificação dos valores na matriz de vínculos e pesos nos apresentou resultados satisfatórios e viabilizou a execução da rota.

4.4 Informações sobre o Trânsito no Cálculo das Rotas

Com a possibilidade de modificar os dados na matriz de vínculos e pesos, temos em (ROSYIDI et al, 2014) a ideia de considerar dados relativos ao trânsito das rodovias, para auxiliar na determinação das rotas.

A Figura 26 mostra um exemplo de uma rede de estradas com suas respectivas distâncias. Nesta rede não temos nenhuma informação sobre o tráfego de veículos entre elas.

Figura 26 – Rede de estradas dependente apenas da distância

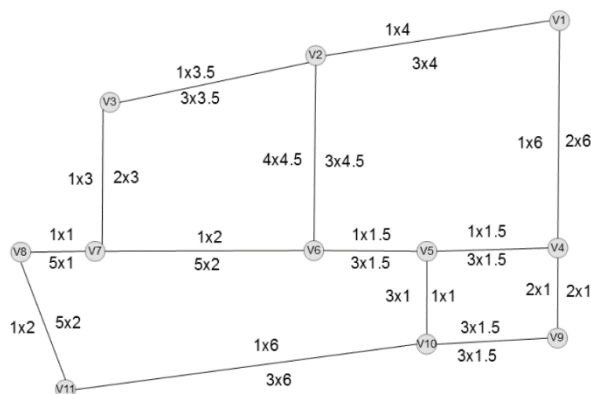


Fonte: Rosyidi et al. (2014)

Classificando as condições do tráfego das estradas em baixo, médio e alto volume de carros, implementa-se um coeficiente que modifica estas distâncias de acordo com as horas do dia.

A Figura 27 apresenta a rede de estradas, nas duas mãos, com as distâncias de cada uma delas multiplicada por um fator de tráfego específico. Este fator varia de acordo com um estudo feito sobre o fluxo de carros nestas estradas em relação às horas do dia.

Figura 27 – Rede de estradas dependente da distância e das condições do tráfego



Fonte: Rosyidi et al. (2014)

Através do algoritmo de Dijkstra, temos as informações de todos os pontos que a rota utilizou até atingir o ponto P0. Pode-se então facilmente determinar que a última conexão utilizada antes de chegar ao seu destino, foi o ponto P1.

Se o carteiro estiver de carro ou caminhão e necessitar partir de P0 para outro ponto afim de entregar mais encomendas, não é desejável permitir que a próxima rota saia de P0 e volte para o ponto P1, obrigando o carteiro a fazer uma manobra de retorno na mesma rua.

Tentando evitar esta manobra, uma vez determinado que o ponto P1 foi o último a ser utilizado antes de completar seu percurso até P0, temporariamente, acrescentamos um peso extra nesta conexão, de modo que o algoritmo de Dijkstra encontre um caminho alternativo para sair de P0 e ir até ao próximo ponto de entrega, que neste exemplo pode ser feito utilizando os pontos P2, P3 e P4.

Embora aparentemente simples, a implementação deste recurso no projeto apresentou bons resultados e rotas factíveis de serem realizadas, sem deixar de considerar também a distância total da rota como o critério principal de avaliação utilizado pelo Algoritmo Genético.

5 RESULTADO E DISCUSSÃO

Para verificação da validade dos mapas gerados pelo software proposto, os mesmos foram comparados com algumas rotas elaboradas pelos carteiros do município em estudo.

Devido à natureza dos algoritmos utilizados, os resultados apresentados podem variar conforme cada execução. Contudo, os valores podem ser considerados satisfatórios ao problema proposto, uma vez que a rota gerada é válida para resolver o problema em questão. Vale destacar que o projeto é um protótipo, logo, o mesmo visa apresentar a capacidade dos algoritmos desenvolvidos em resolver problemas do cotidiano, mesmo aqueles que julgados como ‘dependentes da mão-de-obra humana’. Para uma utilização em massa dos algoritmos é necessário à mudança do software para uma linguagem mais robusta em questão de processamento, uma vez que o problema tratado cresce exponencialmente com a demanda.

Como parâmetros de referência, a tabela 3 nos mostra os dados referentes ao *Hardware* e *Software* utilizados na simulação.

Tabela 3 - Equipamento utilizado na simulação

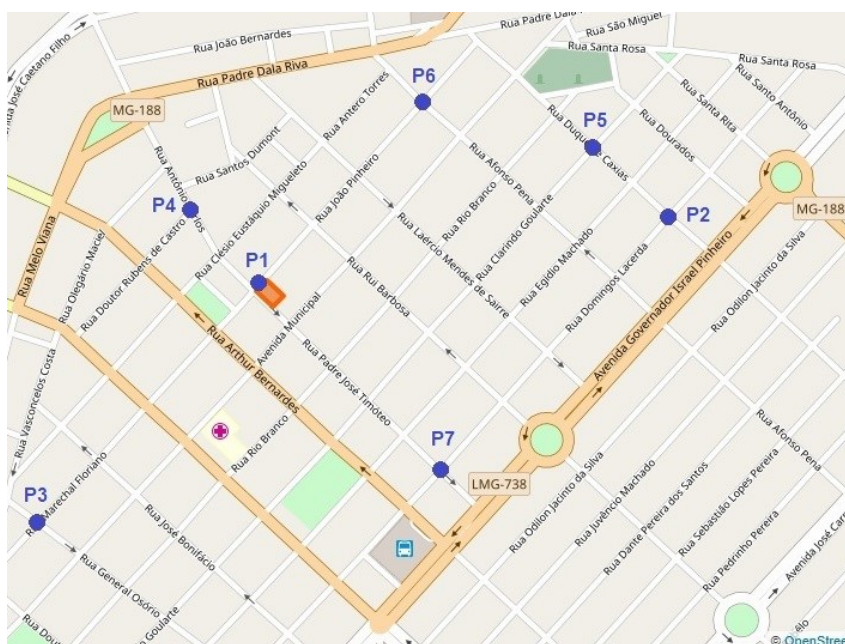
Item	Descrição
Fabricante do Computador:	Samsung Electronics
Processador:	Inter® Core™ i7-3630QM CPU @ 2.40GHz
Memória Instalada (RAM):	6,00GB (Utilizável 5,71GB)
Sistema Operacional:	Windows 10 Home Single Language
Tipo de Sistema:	Sistema Operacional de 64 bits
Software de Simulação	MATLAB R2015a

Fonte: O Autor

5.1 Simulação 1

Como primeiro exemplo, escolhemos seis pontos de entrega distribuídos aleatoriamente no mapa da cidade e definidos como Rota de Entrega. Os pontos foram denominados P₂, P₃, P₄, P₅, P₆ e P₇ respectivamente, sendo reservado o ponto P₁ como sendo o ponto inicial e final da rota de entrega, ou seja, a sede dos correios. Na Figura 29 temos os pontos que o carteiro deverá visitar para entregar as encomendas.

Figura 29 - Pontos de entrega de encomendas – S1



Fonte: OpenStreetMap

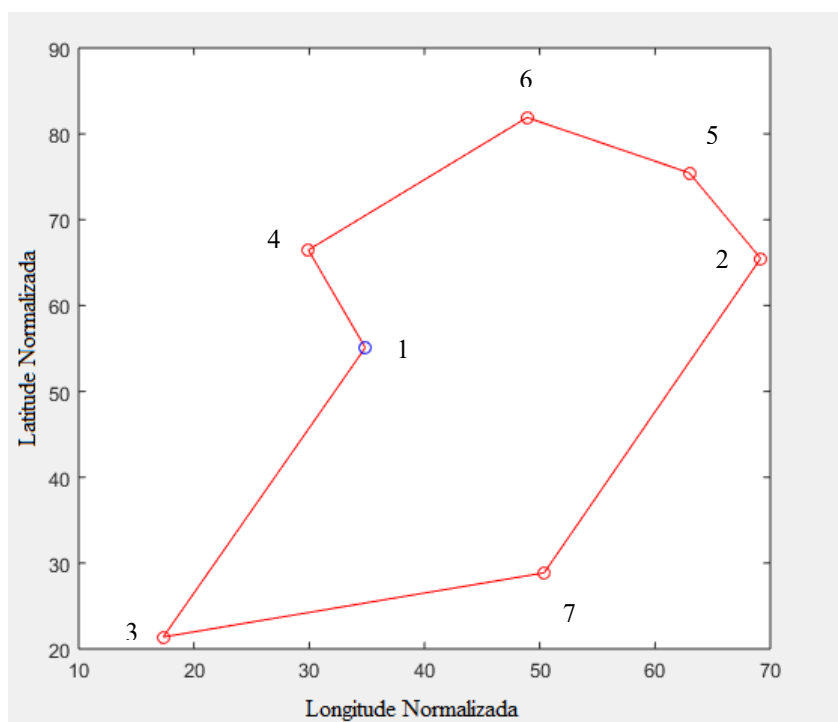
Em seguida executamos o AG para que o mesmo, junto com o Algoritmo de Dijkstra encontre o caminho mais curto para o carteiro efetuar as entregas. Abaixo temos os valores utilizados para a execução do Algoritmo genético:

- Número de Pontos no Cromossomo: 8
- População Inicial: 50
- Gerações: 20
- Função de Avaliação: Menor distância definida pelo Algoritmo de Dijkstra
- Tipo de Cruzamento: *Partially Mapped Crossover* (PMX)
- Probabilidade de Cruzamento: 0.7

- Tipo de Mutação: *Swap Mutation*
- Probabilidade de Mutação: 0.1
- Tipo de Seleção: Elitismo
- Tempo Aproximado de execução: 25 segundos

Uma observação deve ser feita no número de pontos do cromossomo ser igual a 8. A quantidade de pontos de entrega foi definida como 6, mas devemos nos lembrar que o ponto inicial e final do cromossomo ser a própria agência dos correios, que nos dá o total de 8 pontos a serem considerados na criação dos cromossomos. A Figura 30 apresenta o resultado obtido pelo algoritmo.

Figura 30 - Percurso mínimo estipulado pelo AG – S1



Fonte: O Autor

Este resultado apresentado trata-se apenas de uma solução parcial do problema. A prioridade do AG junto com o Algoritmo de Dijkstra é a de encontrar o menor caminho a ser percorrido. Visando aumentar o desempenho do programa, nesta fase inicial, priorizamos o processamento no lugar da apresentação gráfica. A Figura 31 apresenta o cromossomo vencedor e sua respectiva sequência de entrega.

Figura 31 - Cromossomo vencedor

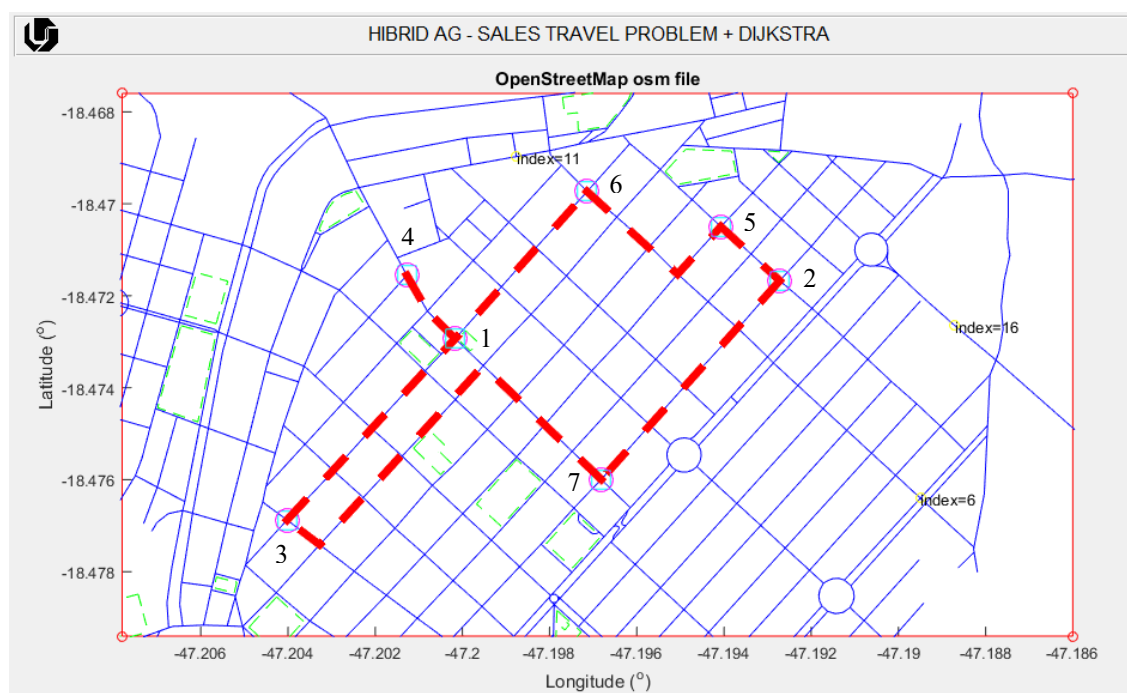
1	4	6	5	2	7	3	1
---	---	---	---	---	---	---	---

Fonte: O autor.

Uma vez determinado o cromossomo vencedor, o programa chega a sua fase final que é a representação gráfica da rota real.

Com base no banco de dados disponibilizado pelo OSM, o mapa do município é desenhado, e o Algoritmo de Dijkstra entra em ação novamente, traçando o caminho ponto a ponto no mapa, conforme apresentado pela Figura 32.

Figura 32 - Rota de Entrega – S1



Fonte: O Autor

É possível observar que o algoritmo obedeceu aos trajetos de um mapa real do município utilizado no estudo, mas a execução do trajeto ficou satisfatória apenas caso o carteiro faça a entrega de bicicleta ou motocicleta.

Note que do ponto P4 ao ponto P6, visando diminuir o trajeto, o algoritmo de Dijkstra utilizou a mesma rua dos pontos P1 e P4, obrigando o entregador a virar e voltar pela mesma rua que chegou. Caso ele esteja de moto ou bicicleta esta manobra é

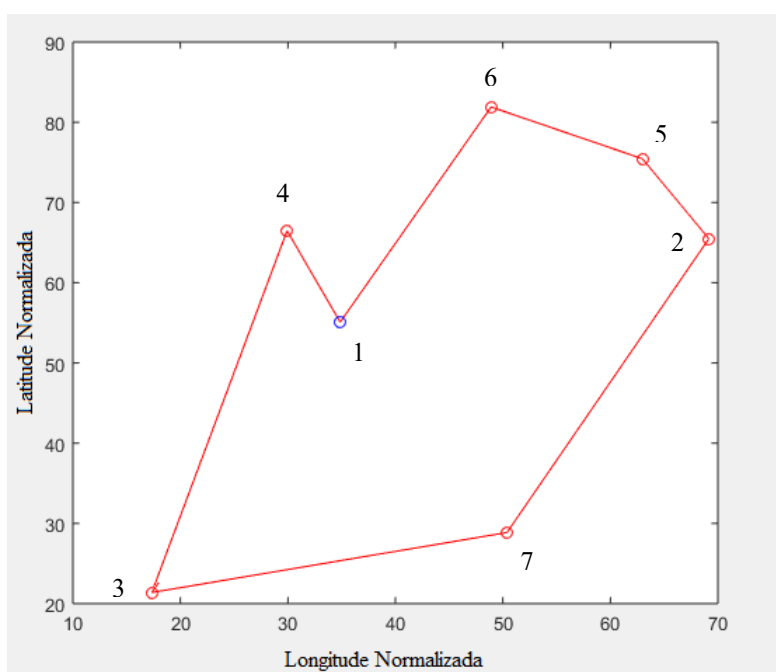
completamente aceitável, uma vez que a rua em questão possui os dois sentidos de circulação, mas considerando que a entrega na maioria das vezes é feita por veículos maiores, esta solução não poderia ser considerada aceitável.

Diante desta constatação, foi criado um recurso no Algoritmo de Dijkstra de modo que o programa possa ter uma opção de roteamento exclusivo para utilização de veículos maiores, evitando que a rota volte pelo mesmo caminho que chegou.

Foi necessário acrescentar um critério de avaliação no momento da elaboração das rotas, incluindo um peso provisório na última conexão antes do ponto de entrega, de modo que o algoritmo de Dijkstra entenda que este caminho não seja uma boa opção a ser utilizada para se deslocar até o próximo ponto de entrega.

Com este novo recurso criado, executamos o AG Híbrido novamente, para que o mesmo encontre o caminho mais curto sem voltar pela mesma rua que chegou. A Figura 33 apresenta o resultado obtido pelo algoritmo.

Figura 33 - Percurso mínimo estipulado pelo AG para Veículos – S1



Fonte: O Autor

Baseado na nova rota, podemos constatar uma pequena modificação no cromossomo vencedor, representado na Figura 34.

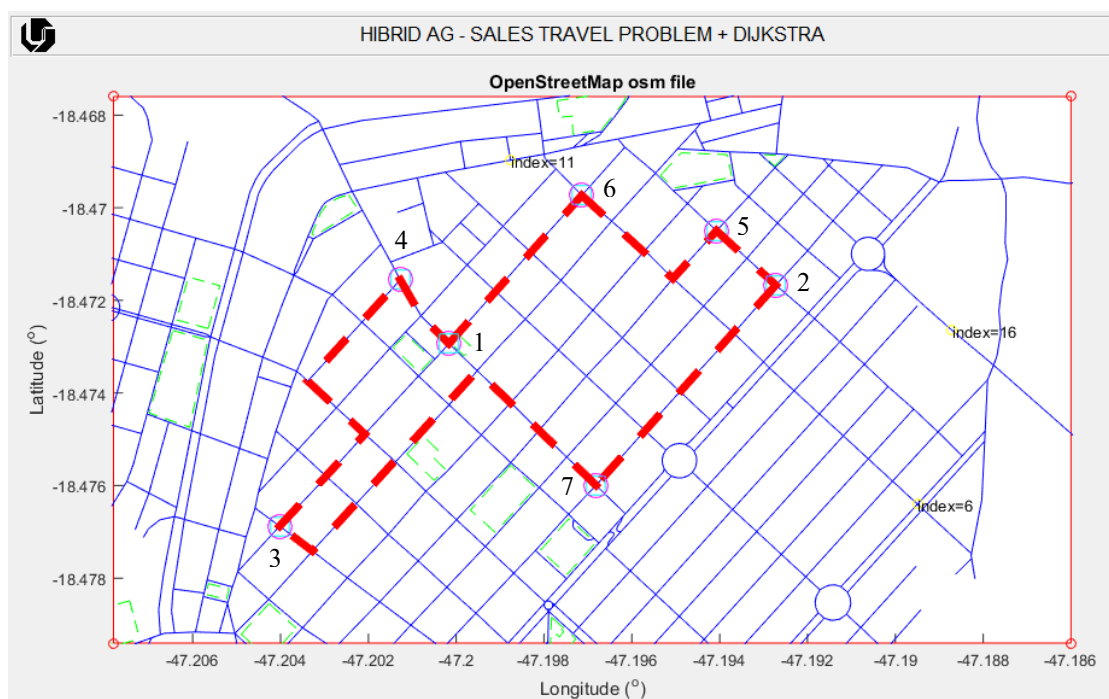
Figura 34 - Cromossomo vencedor S1 - Rota para veículos

1	6	5	2	7	3	4	1
---	---	---	---	---	---	---	---

Fonte: O autor.

Com a nova sequência de entrega disponibilizada pelo AG, executamos Algoritmo de Dijkstra para traçar a nova rota no mapa, apresentada pela Figura 35.

Figura 35 - Rota de entrega para veículos – S1



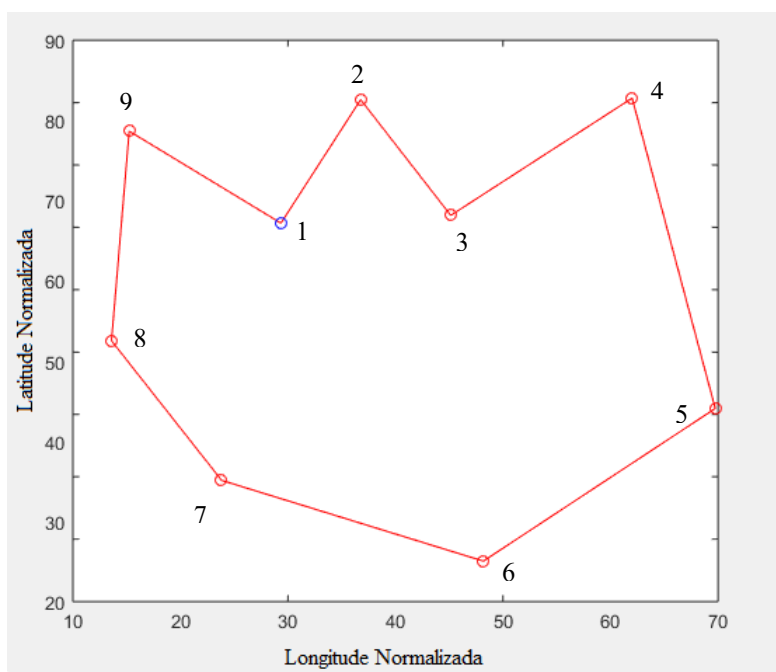
Fonte: O Autor

É possível observar que o algoritmo novamente obedeceu aos trajetos do mapa real do município utilizado no estudo, e que a execução do trajeto ficou satisfatória caso o carteiro faça a entrega de carro, furgão ou até mesmo caminhão.

- Probabilidade de Cruzamento: 0.7
- Tipo de Mutação: *Swap Mutation*
- Probabilidade de Mutação: 0.1
- Tipo de Seleção: Elitismo
- Tempo Aproximado de execução: 45 segundos

A Figura 37 apresenta o resultado obtido pelo algoritmo.

Figura 37 - Percurso mínimo estipulado pelo AG – S2



Fonte: O Autor

A Figura 38 apresenta o cromossomo vencedor

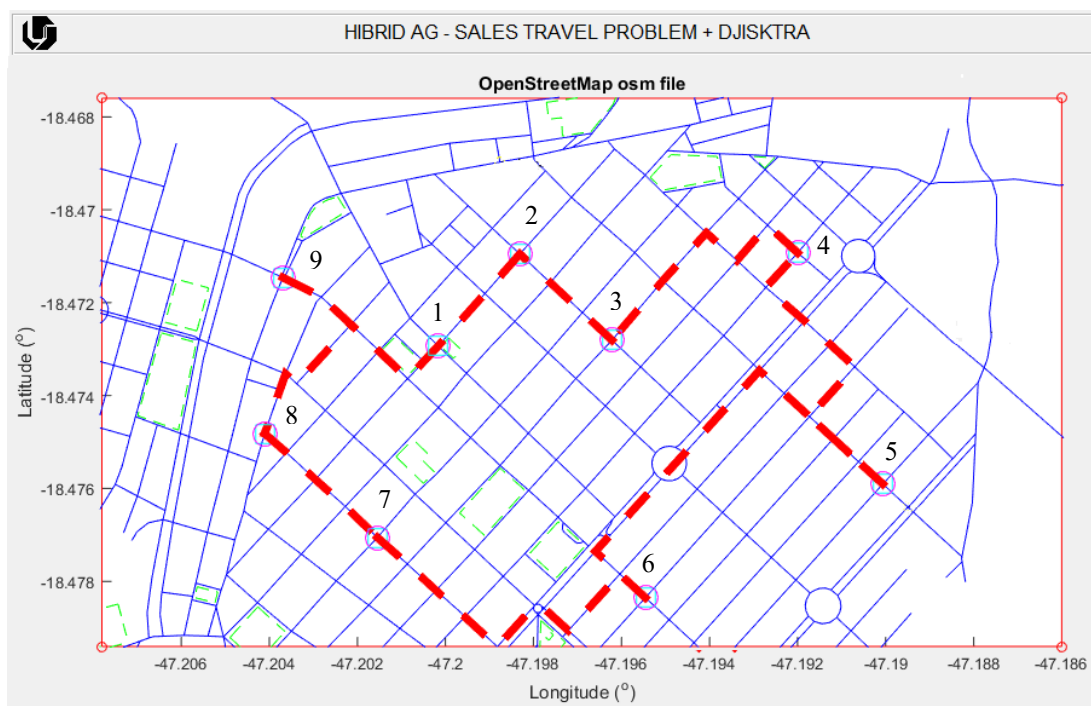
Figura 38 - Cromossomo vencedor – S2

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

Fonte: O Autor

A Figura 39 apresenta a rota da segunda simulação desenhada no mapa da cidade. Podemos observar que esta solução foi obtida com o AG executado sem a restrição de carros e caminhões, ou seja, esta rota é considerada válida apenas para entregas de bicicleta ou motocicleta.

Figura 39 - Rota de Entrega - AG sem restrição S2

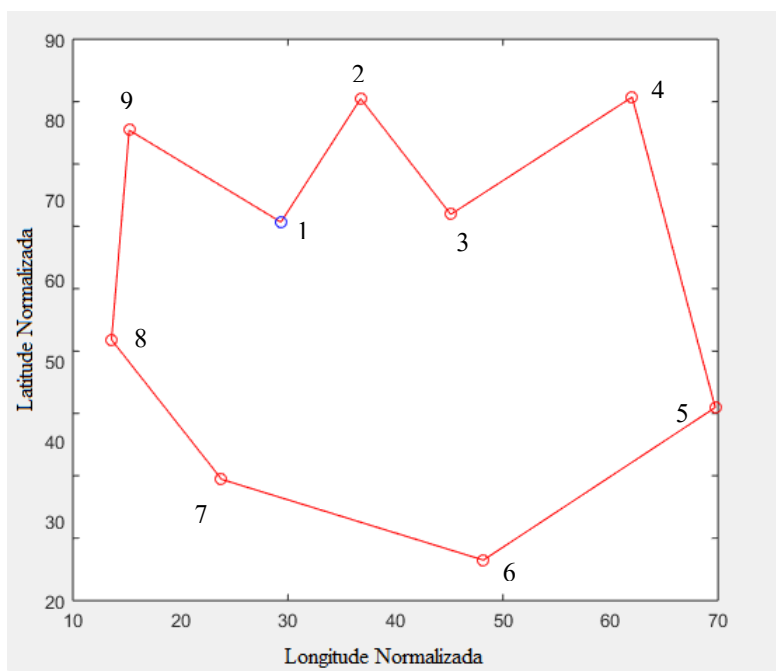


Fonte: O Autor

Note que do ponto P9 ao ponto P8, do ponto P6 ao ponto P5 e do ponto P5 ao ponto P4, o algoritmo de Dijkstra executa uma rota que obriga o entregador a virar e voltar pela mesma rua que chegou. Considerando que a entrega na maioria das vezes é feita por veículos maiores, esta solução também não poderia ser considerada aceitável caso a entrega fosse feita utilizando carro, caminhonete ou caminhão.

Acrescentando-se novamente o critério de avaliação elaborado para veículos e caminhões no momento da elaboração das rotas, critério este que inclui um peso provisório na última conexão antes do ponto de entrega, de modo que o algoritmo de Dijkstra entenda que este caminho não seja uma boa opção a ser utilizada para se deslocar até o próximo ponto de entrega, temos o percurso mínimo estipulado pelo AG, apresentado na Figura 40.

Figura 40 - Percurso mínimo estipulado pelo AG para Veículos – S2



Fonte: O Autor

A Figura 41 apresenta o cromossomo vencedor

Figura 41 - Cromossomo vencedor S2 - Rota para veículos

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

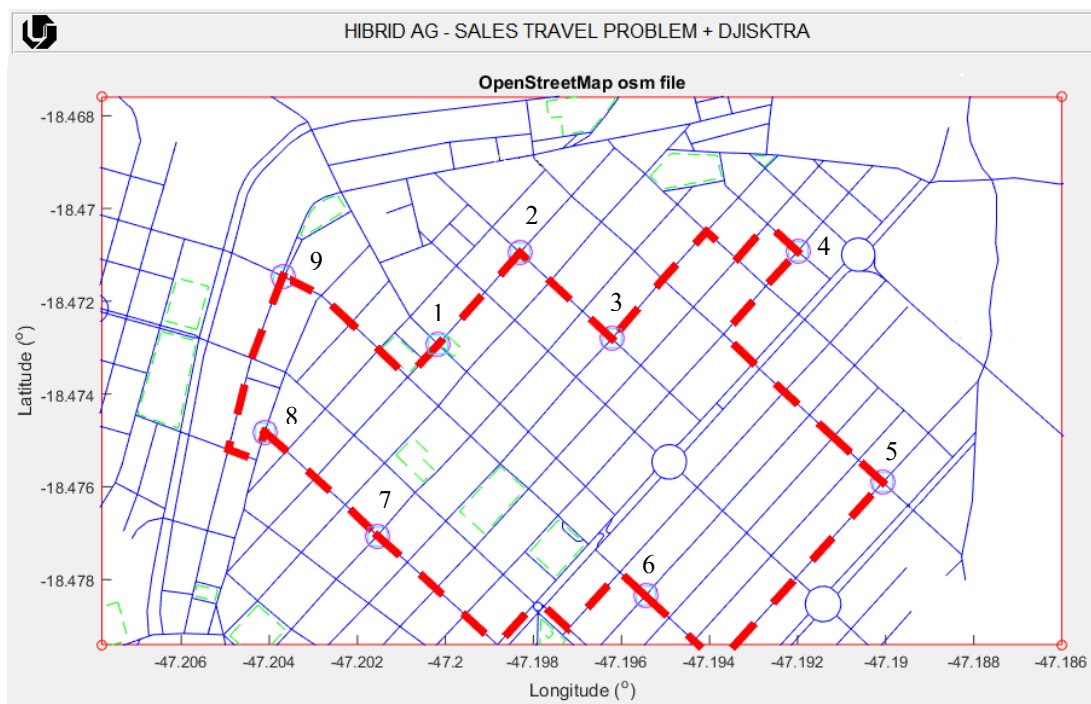
Fonte: O Autor

Analisando os resultados desta simulação, percebe-se que a sequência de entrega do algoritmo executado sem a restrição e a sequência de entrega do algoritmo executado com a restrição para carros e caminhões encontraram a mesma solução.

Da mesma forma que podemos ter sequências de entrega completamente diferentes, quando executamos o algoritmo com e sem a restrição para carros e caminhões, também podemos ter sequências de entrega exatamente iguais, uma vez que o único critério de avaliação a ser considerado durante a execução do algoritmo ser o da menor distância do trajeto.

As sequências de entregas serem iguais não significa que os trajetos para a entrega também serão. Pode-se entender isso facilmente observando o desenho da rota de entrega mostrado pela Figura 42.

Figura 42 - Rota de entrega para veículos – S2



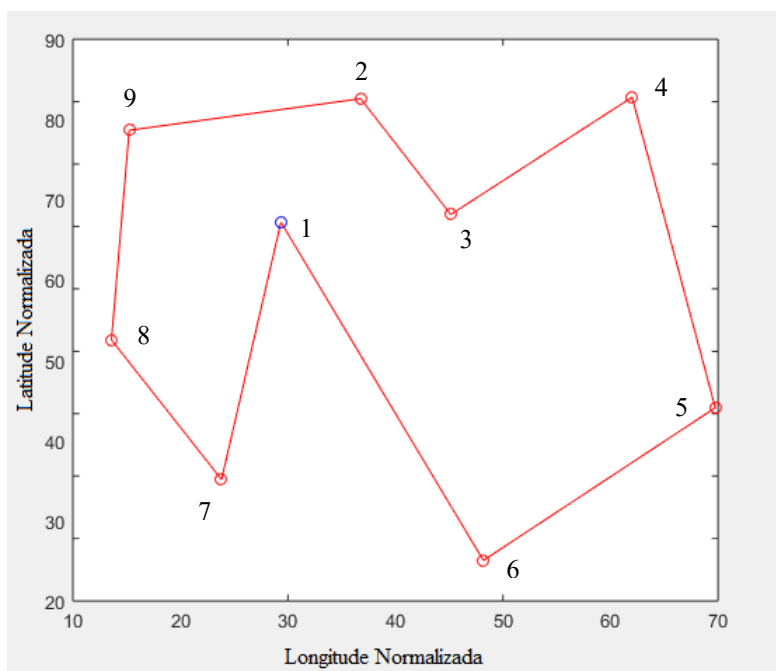
Fonte: O Autor

Observa-se que o algoritmo sempre obedece aos trajetos do mapa real do município e que a execução do trajeto ficou satisfatória caso o carteiro faça a entrega de carro ou caminhão.

Os algoritmos evolucionários são extremamente dependentes de fatores estocásticos (probabilísticos), tanto na fase de inicialização da população quanto na fase de evolução (durante a seleção dos pais, principalmente). Isto faz com que os seus resultados raramente sejam perfeitamente reprodutíveis. Ademais, os algoritmos evolucionários, apesar do seu nome indicar o contrário, são heurísticas que não asseguram a obtenção do melhor resultado possível em todas as suas execuções (LINDEN, 2008).

A Figura 43, apresenta o resultado de uma rota calculada pelo AG no qual a distância ficou um pouco superior ao melhor resultado encontrado nas simulações. Para uma melhor compreensão e didática, denominaremos este resultado como resultado intermediário.

Figura 43 - Resultado intermediário estipulado pelo AG – S2



Fonte: O Autor

A Figura 44 apresenta o cromossomo vencedor do resultado intermediário.

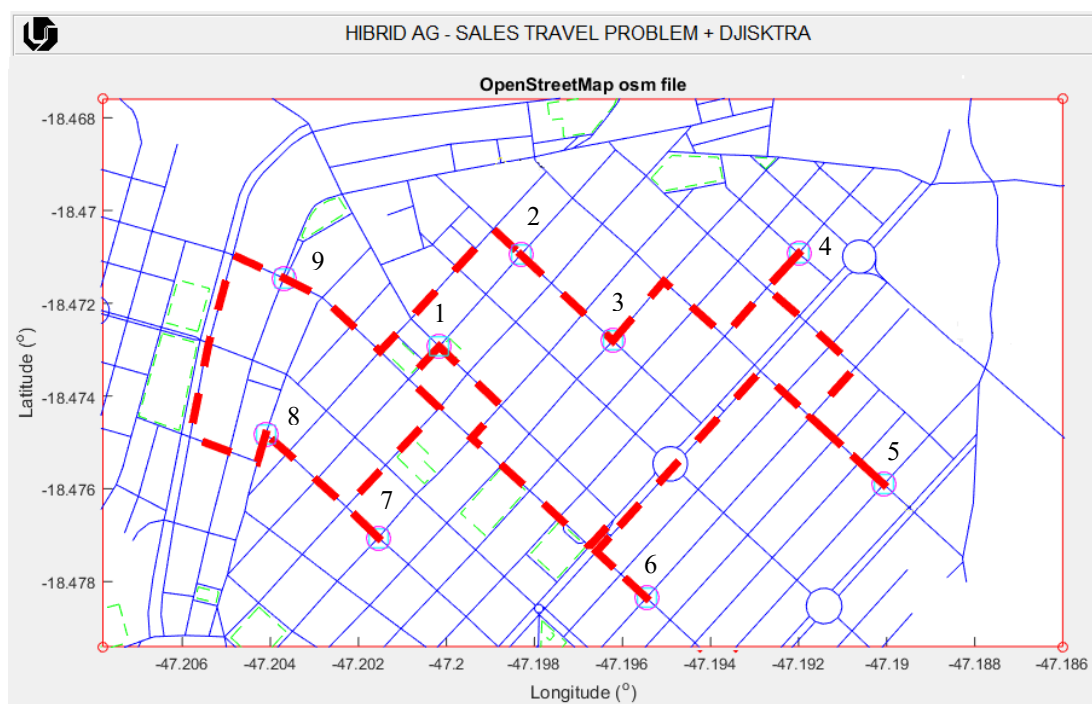
Figura 44 - Cromossomo vencedor – Resultado Intermediário

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

Fonte: O Autor

Na Figura 45 temos a rota de entrega das encomendas, elaborada pelo algoritmo de Dijkstra sem restrição, ideal para entregas de motocicleta ou bicicleta.

Figura 45 - Rota de Entrega Intermediária



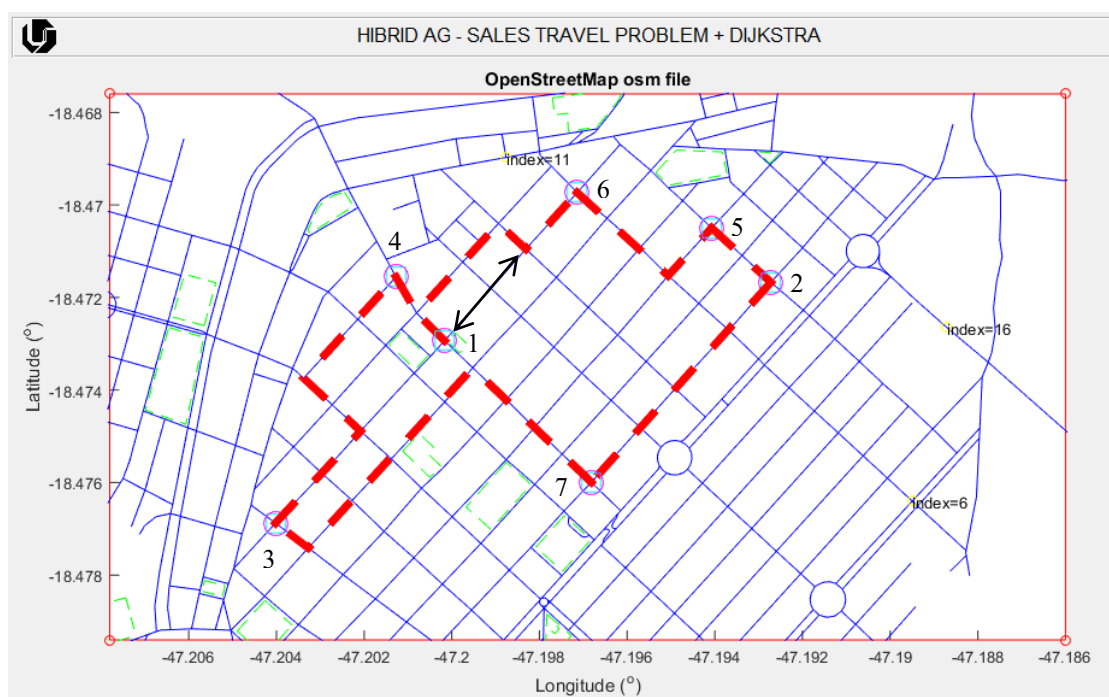
Fonte: O Autor

Diante destes resultados, podemos constatar que, independentemente do local a ser implementado, o AG Híbrido é capaz de gerar uma rota que obedeça aos limites físicos das ruas, quaisquer que sejam os locais de origem ou destino da encomenda.

5.3 Sugestão de Melhoria

Com o desenvolvimento do critério de modificação dos pesos na base de dados das conexões durante a execução do algoritmo, surgiu uma nova proposta que poderá ser desenvolvida em trabalhos futuros, que é a de acrescentar pesos nas conexões de acordo com informações on-line de trânsito. Por não ser o objetivo principal deste estudo, apresentamos apenas como forma de demonstração a mesma rota vencedora apresentada na primeira simulação, com uma simulação de bloqueio de trânsito na rua dos correios (P1), apresentada pela Figura 46.

Figura 46 - Rota Alternativa segundo informações on-line do trânsito



Fonte: O Autor

6 CONCLUSÕES

O trabalho apresentou um método de elaboração de rotas, utilizando o sistema de entrega de encomendas da Empresa de Correios e Telégrafos da cidade de Coromandel-MG como estudo de caso.

Semelhante ao problema do caixeiro viajante, onde o vendedor necessita passar em todas as cidades apenas uma vez, percorrendo a menor distância possível, as dificuldades de elaboração das rotas para entregas das encomendas apresentaram os mesmos desafios para a sua solução. Devido a essa semelhança, definimos a utilização do Algoritmo Genético como estratégia inicial para nossas pesquisas.

Ao compararmos o problema de roteamento das entregas com o problema do caixeiro viajante, constatamos também que se tratava de um problema assimétrico, onde as distâncias de ida e volta dos locais de entrega poderiam variar conforme a disposição das ruas e avenidas da cidade, não podendo ser solucionado com lógica Euclidiana simples. A solução para este problema veio com a escolha do Algoritmo de Dijkstra para calcular as distâncias das rotas elaboradas pelo AG, uma vez que poderíamos representar as casas, ruas e avenidas da cidade em um grafo.

Utilizar o Algoritmo de Dijkstra como meio de avaliação das rotas dos indivíduos gerados pelo Algoritmo Genético resultou ao que denominamos neste estudo de Algoritmo Genético Híbrido.

Os resultados iniciais do algoritmo híbrido criado nos mostraram logo a necessidade de um aprimoramento, visto que poderia haver rotas em que o caminho de ida até o local de entrega e o caminho de volta poderiam ser feitos na mesma rua, podendo ser executado apenas caso o carteiro estivesse de motocicleta ou bicicleta. Fazer o retorno na mesma rua utilizando um automóvel é uma manobra de risco e utilizando um caminhão, na maioria das vezes, seria quase impossível.

Para viabilizar a utilização de automóvel ou caminhão na entrega das encomendas, foi elaborada uma modificação no algoritmo híbrido que possibilitou alterar os pesos na base de dados do grafo criando, de forma temporária, um caminho não desejado ou muito longo, evitando que o Algoritmo de Dijkstra escolhesse o caminho de ida como opção para o caminho de volta.

Com o recurso de alterar os pesos temporariamente na base de dados do grafo e a conexão do algoritmo com algum software de monitoramento de trânsito on-line, poderíamos criar pesos auxiliares que informariam ao Algoritmo de Dijkstra a situação em tempo real das ruas e avenidas, podendo o mesmo criar rotas alternativas, desviado das ruas mais congestionadas.

Na execução do algoritmo, a ação de elaborar todas as rotas possíveis de todos os cromossomos para determinar suas distâncias e assim poder estabelecer o critério de avaliação é a ação que mais exigiu processamento do computador.

Os resultados obtidos foram animadores, e novas pesquisas deverão ser realizadas a fim de melhorar o desempenho da aplicação e facilitar sua adoção nas cadeias logísticas de entrega de encomendas espalhadas pelas diversas cidades do país. Dentre os pontos que deverão ser pesquisados futuramente, destacam-se a utilização do algoritmo em um maior número de pontos de entrega de encomendas e a comparação da qualidade dos resultados obtidos pelo algoritmo frente a outros métodos de estimação de rotas.

REFERÊNCIAS

- BARROS, E. A. R.; PAMBOUKIAN, S. V. D.; ZAMBONI, L. C. Algoritmo de Dijkstra: Apoio Didático e Multidisciplinar na Implementação, Simulação e Utilização Computacional. International Conference on Engineering and Computer Education - 2007
- BORGES, S. K. Resolução de timetabling utilizando algoritmos genéticos e evolução cooperativa. 2003. 103 f. Dissertação (Mestrado em Informática) – Universidade Federal do Paraná, Curitiba, 2003.
- BUENO, M.L.P.; OLIVEIRA, G.M.B. Analyzing the effects of neighborhood crossover in multiobjective multicast flow routing problem. IEEE International Conference on Systems, Man and Cybernetics, 2010. <https://doi.org/10.1109/ICSMC.2010.5641787>.
- CHAKRABORTY, B. Simultaneous search for multiple routes using genetic algorithm. IEEE International Conference on Computational Intelligence for Measurement Systems and Applications, 2004. <https://doi.org/10.1109/CIMSA.2004.1397235>.
- CHAKRABORTY, B.; MAEDA, T.; CHAKRABORTY, G. Multiobjective route selection for car navigation system using genetic algorithm. Proceedings of the 2005 IEEE Midnight-Summer Workshop on Soft Computing in Industrial Applications, 2005. <https://doi.org/10.1109/SMCIA.2005.1466971>.
- CONFEDERAÇÃO NACIONAL DE TRANSPORTES. Pesquisa CNT de rodovias 2017: Relatório Gerencial. 2017. Disponível em: <<http://pesquisarodovias.cnt.org.br/Paginas/relatorio-gerencial>>. Acesso: 28 ago. 2017.
- CORMEN, T. H. et al. *Algoritmos teoria e prática*. 2. ed. Rio de Janeiro: Campus, 2002.
- CORREIOS. História Postal. 2016. Disponível em: <<http://www.correios.com.br/sobre-os-correios/a-empresa/historia/historia-postal/#ECT>>. Acesso em: 10 nov. 2018.
- CUNHA, C. B. Uma contribuição para o problema de roteirização de veículos com restrições operacionais. São Paulo: EPUSP, Departamento de Engenharia de Transportes. 222p. (Tese de Doutorado), 1997.

DIB, O.; MANIER, M.A.; CAMINADA, A. A Hybrid Metaheuristic for Routing in Road Networks. IEEE 18th International Conference on Intelligent Transportation Systems, 2015. <https://doi.org/10.1109/ITSC.2015.129>.

DONG, O. et al. A service routing reconstruction approach in cyber-physical power system based on risk balance. NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, 2018. <https://doi.org/10.1109/NOMS.2018.8406294>.

ELSERSY, M.; ELFOULY, T.M.; AHMED, M.H. Joint Optimal Placement, Routing, and Flow Assignment in Wireless Sensor Networks for Structural Health Monitoring. IEEE Sensors Journal, v. 12, n. 12, 2016. <https://doi.org/10.1109/JSEN.2016.2554462>.

ESTADÃO CONTEÚDO. Comércio eletrônico cresce 7,5% no 1º semestre de 2017, diz EBIT. Revista Pequenas empresas, Grandes Negócios. Disponível em: < [https://revistapegn.globo.com/Negocios/noticia/2017/08/comercio-eletronico-cresce-75-no-1-1-semestre-de-2017-diz-ebit.html](https://revistapegn.globo.com/Negocios/noticia/2017/08/comercio-eletronico-cresce-75-no-1-1-1-semestre-de-2017-diz-ebit.html) >. Acesso em: 26 ago. 2018.

FERARIU, L.; CÎMPANU, C. Pareto genetic path planning hybridized with multi-objective Dijkstra's algorithm. 18th International Conference on System Theory, Control and Computing (ICSTCC), 2014. <https://doi.org/10.1109/ICSTCC.2014.6982439>.

GOLDBARG, M. C.; LUNA, H. P. L. Otimização Combinatória e Programação Linear. Rio de Janeiro: CAMPUS, 2000.

HASAN, B.S; KHAMEES, M.A.; MAHMOUD, A.S.H. A Heuristic Genetic Algorithm for the Single Source Shortest Path Problem. IEEE/ACS International Conference on Computer Systems and Applications, 2007. <https://doi.org/10.1109/AICCSA.2007.370882>.

HELGAUN, K. An effective implementation of the Lin-Kernighan Traveling Salesman Heuristic, European Journal of Operational Research, v.126, p.106-130, 2000. [https://doi.org/10.1016/S0377-2217\(99\)00284-2](https://doi.org/10.1016/S0377-2217(99)00284-2)

HUANG, M. et al. Immune Algorithm Based Routing Optimization in Fourth-Party Logistics. IEEE International Conference on Evolutionary Computation, 2006. <https://doi.org/10.1109/CEC.2006.1688691>.

INAGAKI, J.; HASEYAMA, M.; KITAJIMA, H. A GA-based routing method with an upper bound constraint. Proceedings of the 2003 International Symposium on Circuits and Systems, v. 2, 2003. <https://doi.org/10.1109/ISCAS.2003.1206016>.

INAGAKI, J.; HASEYAMA, M.; KITAJIMA, H. A genetic algorithm for determining multiple routes and its applications. ISCAS'99. Proceedings of the 1999 IEEE International Symposium on Circuits and Systems VLSI (Cat. No.99CH36349), v. 6, 1999. <https://doi.org/10.1109/ISCAS.1999.780114>.

INAGAKI, J.; HASEYAMA, M.; KITAJIMA, H. A new genetic algorithm for routing the shortest route via several designated points. IEEE International Symposium on Circuits and Systems (Cat. No.01CH37196), v. 2, 2001. <https://doi.org/10.1109/ISCAS.2001.921046>.

JINZHU, M.; JIANGUO, W.; QUN, E. Mission planning of ship-to-ship missile with jamming load on the penetration. Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference, 2014. <https://doi.org/10.1109/CGNCC.2014.7007279>.

JUNG, S.; ARIYUR, K.B. Robustness for large scale UAV autonomous operations. IEEE International Systems Conference, 2011. <https://doi.org/10.1109/SYSCON.2011.5929115>.

JUZOJI, H.; NAKAJIMA, I.; KITANO, T. A Development of Network Topology of Wireless Packet Communications for Disaster Situation with Genetic Algorithms or with Dijkstra's. 2011 IEEE International Conference on Communications (ICC), 2011. <https://doi.org/10.1109/icc.2011.5962439>.

KAMBAYASHI, Y. et al. Dijkstra beats genetic algorithm: Integrating uncomfortable intersection-turns to subjectively optimal route selection. IEEE International Conference on Computational Cybernetics (ICCC), 2009. <https://doi.org/10.1109/ICCCYB.2009.5393932>.

KAMEL, Z.; SLIM, H. CGOMFP: control genetic operators with management of the final population to optimize a multimodal transport moving. IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583), v. 7, 2004. <https://doi.org/10.1109/ICSMC.2004.1401375>.

LAPORTE, G. The Vehicle-Routing Problem – An Overview of exact and approximate algorithms, European Journal of Operational Research, Vol. 59, p. 345-358, 1992.

[https://doi.org/10.1016/0377-2217\(92\)90192-C](https://doi.org/10.1016/0377-2217(92)90192-C)

LENSTRA, J. & RINNOOY KAN, A. Complexity of vehicle routing and scheduling problems, Networks, Vol.11, p. 221-228, 1981. <https://doi.org/10.1002/net.3230110211>.

LI, W.Y.; CHEN, X.W.; YANG, B. Bus Travel Transit Path Query Algorithm Based on Ant Algorithm. Third International Conference on Genetic and Evolutionary Computing, 2009. <https://doi.org/10.1109/WGEC.2009.13>.

LIN, C.H, et al. Genetic Algorithm for Shortest Driving Time in Intelligent Transportation Systems. International Conference on Multimedia and Ubiquitous Engineering, 2008. <https://doi.org/10.1109/MUE.2008.16>.

LINDEN, R. Algoritmos Genéticos: uma importante ferramenta da inteligência computacional. Rio de Janeiro: Brasport, 2008.

LIU, Q.; WU, C. A novel selection approach for replicated multicast servers using genetic algorithm. International Conference on Wireless Communications, Networking and Mobile Computing, v. 2, 2005. <https://doi.org/10.1109/WCNM.2005.1544207>.

LIU, Y. et al. A Genetic Algorithm for Solving the Bandwidth-Constrained Multi-path Problem. 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence, 2015. <https://doi.org/10.1109/ACIT-CSI.2015.96>.

MAIA, F.; BERNARDES, A. Brasil gasta o equivalente a 11,6% do PIB com logística em rodovias: as causas para o alto custo logístico das mercadorias brasileiras esbarram na infraestrutura que contribui diretamente para o custo Brasil. Correio Braziliense – Cidades. 2016. Disponível em: < https://www.correiobraziliense.com.br/app/noticia/cidades/2016/08/11/interna_cidadesdf,544003/brasil-gasta-o-equivalente-a-11-6-do-pib-com-logistica.shtml> Acesso em: 21 nov. 2018.

MCMAHON, G.; SEPTIAWAN, R.; SUGDEN, S. A multiservice traffic allocation model for LEO satellite communication networks. IEEE Journal on Selected Areas in Communications, v. 22, n. 3, 2004. <https://doi.org/10.1109/JSAC.2004.823417>.

MÉNDEZ, Y. S.; GUARDIA, L. E. T. Problema do Caminho Mais Curto: Algoritmo de Dijkstra. 2008. 12f. Artigo (Mestrado em Engenharia de Produção) Universidade Federal Fluminense, Niterói, 2008.

MORINAGA, E. et al. A facility layout planning method considering routing and temporal efficiency. International Symposium on Flexible Automation (ISFA), 2016. <https://doi.org/10.1109/ISFA.2016.7790158>.

MURUGANANTHAM, N.; EL-OCLA, H. Genetic Algorithm-Based Routing Performance Enhancement in Wireless Sensor Networks. IEEE 3rd International Conference on Communication and Information Systems (ICCIS), 2018. <https://doi.org/10.1109/ICOMIS.2018.8644725>.

NANAYAKKARA, S.C. et al. Genetic Algorithm based route planner for large urban street networks. IEEE Congress on Evolutionary Computation, 2007. <https://doi.org/10.1109/CEC.2007.4425056>.

OPEN STREET MAP. O Open Street Map fornece dados a centenas de sites na internet, aplicações de celular e outros dispositivos. 2016. Disponível em: <<http://www.openstreetmap.org/about>>. Acesso em: 01 nov. 2016.

PODDAR, R; BANERJEE, S.; THAKURTA, P.K.G. Shortest Path Routing for Coordinate Based Mobile Networks: A Genetic Algorithm Approach. International Conference on Advances in Mobile Network, Communication and Its Applications, 2012. DOI: 10.1109/MNCApps.2012.15. <https://doi.org/10.1109/MNCApps.2012.15>.

REINELT, G. The Traveling Salesman – Computational Solutions for TSP Applications. Berlin: Springer-Verlag, 1994.

RENAUD, J. & BOCTOR, F. F. A sweep-based algorithm for the fleet size and mix vehicle routing problem, European Journal of Operational Research, Vol. 140, p. 618-628, 2002. [https://doi.org/10.1016/S0377-2217\(01\)00237-5](https://doi.org/10.1016/S0377-2217(01)00237-5).

ROCHA, A. M. Resolução do problema do caixeiro viajante assimétrico (e uma variante) através da relaxação Lagrangeana. Universidade do Minho – Portugal, 2006.

RODRIGUES, G. P. Otimização de rotas através da aplicação de algoritmos exatos e heurísticos. 2004. 53 f. Monografia (Graduação EM Ciência da Computação), Universidade Presidente Antônio Carlos, Barbacena, 2004.

ROSYIDI, L. et al. Timebase Dynamic Weight for Dijkstra Algorithm Implementation in Route Planning Software. International Conference on Intelligent Green Building and Smart Grid (IGBSG), 2014. <https://doi.org/10.1109/IGBSG.2014.6835261>.

SANTOS, F.A.; MATEUS, G.R. A Hybrid Evolutionary Multi-Objective Algorithm to Setup Explicit Routes in MPLS Networks. Eighth International Conference on Hybrid Intelligent Systems, 2008. <https://doi.org/10.1109/HIS.2008.165>.

SHAKSHUKI, E.; XING, X.; MALIK, H. Mobile Agent for Efficient Routing among Source Nodes in Wireless Sensor Networks. Third International Conference on Autonomic and Autonomous Systems (ICAS'07), 2007.
<https://doi.org/10.1109/CONIELECOMP.2007.83>.

SHAO, S.; GUAN, W.; BI, J. Electric vehicle-routing problem with charging demands and energy consumption. IET Intelligent Transport Systems, v. 12, n. 3, 2018.
<https://doi.org/10.1049/iet-its.2017.0008>.

SILVEIRA, J. F. P. O problema do caixeiro viajante. Porto Alegre: UFRGS. 2000.). Disponível em: <<http://www.mat.ufrgs.br/~portosil/caixeiro.html>>. Acesso em: 03 nov. 2016.

TAKÁO, T.B. et al. A hybrid intelligent Body Sensor Networks model. 11th International Conference on Hybrid Intelligent Systems (HIS), 2011.
<https://doi.org/10.1109/HIS.2011.6122082>.

VIEIRA, H. P. Metaheurística para a solução de problemas de roteamento de veículos com janela de tempo. 2013, 108f. Dissertação (Mestrado em Matemática Aplicada) – Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas, Campinas, 2013.

YOSHIKAWA, M.; TERA, H. Architecture of virus evolutionary genetic algorithm for car navigation system. 50th Midwest Symposium on Circuits and Systems, 2007. <https://doi.org/10.1109/MWSCAS.2007.4488717>.

YOSHIKAWA, M.; TERA, H. Car Navigation System Based on Hybrid Genetic Algorithm. WRI World Congress on Computer Science and Information Engineering, v. 5, 2009. <https://doi.org/10.1109/CSIE.2009.558>.

ZAGHDOUD, R. et al. A Multi-objective Approach for Assignment Containers to AIVs in a Container Terminal. IEEE International Conference on Systems, Man, and Cybernetics, 2013. <https://doi.org/10.1109/SMC.2013.420>.

ZAMBONI, L. C.; MONEZZI JÚNIOR, O., “Cálculo Numérico para Universitários”. São Paulo: Páginas & Letras, 2002.