

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE – UERN
FACULDADE DE CIÊNCIAS EXATAS E NATURAIS – FANAT
DEPARTAMENTO DE INFORMÁTICA – DI

SEBASTIÃO MATEUS MARQUES DE MENEZES

**OTIMIZAÇÃO DE ROTEAMENTO DE VEÍCULOS PARA COLETA DE LIXO DE
DESCARTE IRREGULAR EM ÁREA PÚBLICA**

MOSSORÓ - RN

2019

SEBASTIÃO MATEUS MARQUES DE MENEZES

**OTIMIZAÇÃO DE ROTEAMENTO DE VEÍCULOS PARA COLETA DE LIXO DE
DESCARTE IRREGULAR EM ÁREA PÚBLICA**

Monografia apresentada à Universidade do Estado do Rio Grande do Norte como um dos pré-requisitos para obtenção do grau de bacharel em Ciência da Computação, sob orientação da Prof. Dr. Francisco Chagas de Lima Júnior.

MOSSORÓ - RN

2019

**Catálogo da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.**

M543o Menezes, Sebastião Mateus Marques de
Otimização de Roteamento de Veículos para coleta de
lixo de descarte irregular em área pública. / Sebastião
Mateus Marques de Menezes. - Mossoró, RN, 2019.
52p.

Orientador(a): Prof. Dr. Francisco Chagas de Lima
Júnior.

Coorientador(a): Profa. M^a. Elisa de Fátima Andrade
Soares.

Monografia (Graduação em Ciência de Computação).
Universidade do Estado do Rio Grande do Norte.

1. Roteamento de Veículos. 2. Coleta de Lixo. 3.
Otimização Combinatória. 4. Descarte Irregular. I. Lima
Júnior, Francisco Chagas de. II. Universidade do Estado
do Rio Grande do Norte. III. Título.

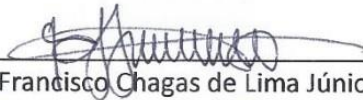
SEBASTIÃO MATEUS MARQUES DE MENEZES

OTIMIZAÇÃO DE ROTEAMENTO DE VEÍCULOS PARA COLETA DE LIXO DE DESCARTE IRREGULAR EM ÁREA PÚBLICA

Monografia apresentada como pré-requisito para a obtenção do título de Bacharel em Ciência da Computação da Universidade do Estado do Rio Grande do Norte – UERN, submetida à aprovação da banca examinadora composta pelos seguintes membros:

Aprovada em: 13/05/2019

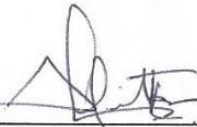
Banca Examinadora



Prof. Dr. Francisco Chagas de Lima Júnior - (Orientador)
Universidade do Estado do Rio Grande do Norte - UERN



Elisa de Fátima Andrade Soares - (Coorientadora)
Universidade do Estado do Rio Grande do Norte - UERN



Prof. Dr. Carlos Heitor Pereira Liberalino - (Examinador)
Universidade do Estado do Rio Grande do Norte - UERN



Prof. Dr. Isaac de Lima Oliveira Filho - (Examinador)
Universidade do Estado do Rio Grande do Norte - UERN

AGRADECIMENTOS

Gostaria de começar agradecendo a toda minha família que sempre acreditou em mim, e principalmente a minha mãe e meu pai e meus irmãos, pois, tudo que eu faço é para eles.

Agradeço a todos os professores da universidade por todo o conhecimento passado durante o curso, principalmente ao professor Lima Júnior por ser meu orientador neste trabalho e em projetos de pesquisa e por me apresentar a área de otimização. Também aos professores Isaac e Heitor que aceitaram compor a banca examinadora.

Agradeço a todos os meus amigos do curso, em especial a Elisa que também foi minha coorientadora e me ajudou muito durante o curso e a André, Alfredo, Davi, Vítor e Veridiano que foram meus parceiros desde o começo do curso, ajudando sempre em tudo que eu precisava e me fazendo rir. Espero que essa parceria não acabe nunca.

Agradeço também aos meus amigos Edjaedson e Juninho por sempre me apoiarem e me motivarem e por estarem comigo em todas as situações, sem vocês nada disso seria possível.

“Nunca esqueça o que você é, porque certamente o resto do mundo não esquecerá. Faça disto sua força, assim nunca poderá ser sua fraqueza. Arme-se nesta lembrança, desta forma ela nunca será usada para ferir você”.

George R. R. Martin

RESUMO

O crescimento populacional nos centros urbanos de médio e grande porte aliado ao aumento de consumo de bens descartáveis tem como consequência a amplificação da produção de lixo. Em virtude dessa produção de lixo composta principalmente por resíduos sólidos, nas áreas urbanas surgem problemas relativos à existência de locais inadequados para o descarte desses resíduos. Apesar do investimento público em coleta seletiva, tratamento apropriado, criação de aterros sanitários dentre outras coisas, ainda é comum em cidades desenvolvidas encontrar locais irregulares como: canteiros, ruas desertas, terrenos baldios, etc. que são utilizados para descarte irregular do lixo. Este tipo de comportamento pode acarretar em sérios problemas, incluindo o surgimento de doenças e a agravação de enchentes. Diante dessa situação, este trabalho tem como objetivo desenvolver um método de otimização de roteamento de veículos para a coleta de lixo de descarte irregular em área pública para minimizar os custos gastos com a etapa de coleta. Tem como objetivos específicos desenvolver um método de otimização que utiliza algoritmos heurísticos para oferecer uma rota com objetivo de minimizar os custos na realização da coleta e validar os algoritmos com dados reais da cidade de Mossoró-RN.

Palavras-chave: Roteamento de veículos, Coleta de Lixo, Otimização Combinatória, Descarte Irregular.

ABSTRACT

The population growth in the mediums and larges urban centers allied with the raise in the consuming of disposable products has the amplification of waste production as a consequence. Due to this production of garbage composed mainly of solid waste, in the urban areas a problem of appropriate places to discard arise. Although the public investment on selective collection, suitable treatment, creation of landfills, and other things, it still is common in developed cities to find innapropriate places like: deserted streets, abandoned buildings, etc. that are used to improper disposable of solid waste. This type of behavior may have serious issues, including the emergence of diseases and aggravation of floods. Against the foregoing, this work aims to develop a optimization method based on vehicle routing to the collection of improper dispose of waste in public areas to minimize the costs with the collection step. Its specific objectives are develop a optimization method using heuristic algorithms to offer a route that minimizes the costs in the collection and validate the algorithms with real data from the city of Mossoró-RN

Keywords: Vehicle Routing, Waste Collection, Combinatorial Optimization, Improper Disposable.

LISTA DE SIGLAS

ABNT	Associação Brasileira de Normas Técnicas.
ABRELPE Pública e Resíduos Especiais.	Associação Brasileira de Empresas de Limpeza
AG	Algoritmos Genéticos.
ALNS	Adaptative Large Neighborhood Search.
API	Application Programming Interface.
CVRP	Capacitated Vehicle Routing Problem.
DCVRP	Distance-Constrained Vehicle Routing Problem.
GIRSU Urbanos.	Gerenciamento Integrado de Resíduos Sólidos
GPS	Global Positioning System.
GRASP	Greedy Randomized Adaptive Search Procedure.
HTTP	Hypertext Transfer Protocol.
IBGE	Instituto Brasileiro de Geografia e Estatística.
JSON	JavaScript Object Notation.
LC	Lista de Candidatos
LRC	Lista Restrita de Candidatos.
NBR	Norma Brasileira.
PMC	Problema das Medianas Capacitado.
PMNC	Problema das Medianas Sem Custo.
REST	Representational State Transfer.
RS	Resíduos Sólidos.
RSS	Resíduos de Serviços de Saúde.
RSU	Resíduos Sólidos Urbanos.
SIG	Sistema de Informações Geográficas.
VNS	Variable Neighborhood Search.

VRP	Vehicle Routing Problem.
VRPPD	Vehicle Routing Problem with Pickup and Delivery.
VRPTW	Vehicle Routing Problem with Backhauls.
VRPTW	Vehicle Routing Problem with Time Window.

LISTA DE FIGURAS

Figura 1 – Etapas integradas de um GIRSU.....	18
Figura 2 – Variações básicas do VRP.....	22
Figura 3 – Uma resolução do problema das p-medianas.....	23
Figura 4 – Representação gráfica da implementação de um cruzamento.....	28
Figura 5 - As etapas do algoritmo desenvolvido.....	33
Figura 6 - Representação de um indivíduo.....	35
Figura 7 - Resultado de um cruzamento.....	36
Figura 8 - Representação de uma mutação.....	36
Figura 9 - Influência do valor de alfa.....	38
Figura 10 - Imagem gerada com resultado de uma rota.....	39
Figura 11 – Pontos de descarte irregular no bairro Belo Horizonte.....	42
Figura 12 – Instância do bairro Belo Horizonte com duas medianas.....	43
Figura 13 – Instância do bairro Belo Horizonte com três medianas.....	44
Figura 14 – Imagem gerada pelo algoritmo com a primeira rota da primeira execução.....	45
Figura 15 – Imagem gerada pelo algoritmo com a segunda rota da primeira execução.....	45
Figura 16 – Imagem gerada pelo algoritmo com a primeira rota da segunda execução.....	46
Figura 17 – Imagem gerada pelo algoritmo com a segunda rota da segunda execução.....	46

SUMÁRIO

1. INTRODUÇÃO	13
2. REFERENCIAL TEÓRICO.....	16
2.1 Resíduos Sólidos	16
2.2 Otimização Combinatória	19
2.3 Problemas de Roteamento de Veículos.....	20
2.4 Problemas das p-medianas.....	22
2.5 Heurísticas e Metaheurísticas.....	23
2.6 GRASP	25
2.6.1 Construção da Solução	26
2.6.2 Busca Local	26
2.7 Algoritmos Genéticos	26
2.7.1 Estrutura de um Algoritmo Genético	27
2.7.2 Cruzamento	27
2.7.2 Mutação	28
2.7.3 Seleção	29
2.8 Trabalhos Relacionados	29
3. IMPLEMENTAÇÃO.....	31
3.1 Estrutura Geral.....	32
3.1.1 Transformador de Matrizes.....	33
3.1.2 Algoritmo Genético.....	34
3.1.3 GRASP.....	37
3.1.4 Gerador de Mapas.....	38
3.1.5 Programa Principal.....	40
4. VALIDAÇÃO E RESULTADOS.....	41
4.1 Metodologia dos testes	41
4.2 Resultados	41
5. CONSIDERAÇÕES FINAIS.....	48
REFERÊNCIAS.....	50

1. INTRODUÇÃO

Atualmente a maioria das pessoas habitam nos centros urbanos. Segundo o último censo feito pelo Instituto Brasileiro de Geografia e Estatística (IBGE) realizado em 2010, constatou-se que 84,4% da população residem em áreas urbanas. Essa expansão está correlacionada com o crescimento populacional bem como o aumento do consumo de bens descartáveis resultando em grandes quantidades de lixo, que tem sido descartado de maneira incorreta, acarretando em várias consequências. Entre elas se destaca o descarte realizado em locais inadequados. Essa prática tem causado o surgimento de áreas inapropriadas nas cidades para o descarte de lixo, gerando problemas aos gestores públicos, assim como risco a saúde das pessoas e contaminação dos solos.

Segundo a Associação Brasileira de Empresas de Limpeza Pública e Resíduos Especiais (ABRELPE) no ano de 2016 cerca de 2692 cidades depositavam a maior parte do resíduo urbano em lugares irregulares. Esses lixões, ao passarem pelo processo de degradação, prejudicam o meio ambiente e podem contaminar a saúde humana em caso de grande exposição. Conforme Cavalcante e Franco (2007) essa contaminação acontece de duas formas principais: Contato direto com agentes patogênicos (físicos, químicos ou biológicos) e também por meio da amplificação de fatores de risco, se tornando uma ameaça ocupacional, ambiental ou alimentar.

Além das doenças, esse tipo de empilhamento de resíduos pode causar sérios problemas ambientais, tais como contaminação do ar e agravamento de enchentes, devido aos entupimentos de esgotos e assoreamento de rios. Essa situação está diretamente relacionada com a falta de consciência ambiental de grande parte da população, que por mais que tenha ciência do perigo que os resíduos mal descartados podem trazer, não realizam a maneira correta e prosseguem poluindo os lugares públicos, assim contribuem para o crescimento do número de lixões em locais distintos das cidades. Em agosto de 2014 uma pesquisa realizada pela ABRELPE revelou que esse número já chega em 2976.

Entretanto esse número não deveria ser tão alto, visto que a lei 12305, da Política Nacional de Resíduos Sólidos determinava um prazo até agosto de 2014 para o fechamento de todos os lixões do Brasil, onde o lixo deveria ser direcionado para

aterros sanitários e ter os devidos cuidados. Além disso, existem áreas com concentrações menores de lixo onde geralmente a população de uma rua ou bairro deposita os resíduos em canteiros, terrenos baldios, prédios abandonados, etc. Este tipo de descarte, na maioria das vezes, é ignorado pelos serviços públicos de coleta de lixo.

Como forma de amenizar a situação, faz-se necessário a criação de programas e campanhas voltados a conscientização da população sobre os perigos, além de mostrar novas alternativas como, a coleta seletiva e a reciclagem. Entretanto, a situação atual é bem preocupante pelo motivo da grande quantidade já existente de lixo descartado. Conforme D'almeida et al. (2018), os serviços de limpeza absorvem aproximadamente entre 5 e 15% dos recursos do orçamento de uma cidade, dos quais 40 a 60% são destinados exclusivamente à coleta e ao transporte de lixo. Posto isso, faz-se necessário que os governantes invistam em formas inteligentes de coletar os resíduos produzidos, objetivando diminuir os custos além de melhorar a qualidade de vida da população.

A produção de lixo nas cidades é inevitável e no Brasil isso não é diferente. A ABRELPE revelou em 2014 que cada brasileiro produziu uma média de 383 quilos de lixo por ano. Essa geração acontece de forma variada, dependendo do nível de desenvolvimento econômico da população. É de competência de cada município tratar dos sistemas de limpeza urbana depois de uma análise da situação, adotando a melhor estratégia para o gerenciamento integrado do lixo. Existem alguns fatores que dificultam esta tarefa e que não devem ser tratados passivamente, pois, na falta de cuidado, é provável que hajam resultados desastrosos. D'almeida et al (2018) destacam algumas dessas dificuldades, como:

- Limitações financeiras;
- Deficiência na capacitação técnica e profissional;
- Descontinuidade política e administrativa;
- Ausência de controle ambiental.

Dada a situação atual de várias cidades pelo Brasil, é imprescindível que este assunto seja tratado mais rigorosamente e que seja adotada alguma estratégia a fim de melhor direcionar os recursos financeiros da área da limpeza urbana no caminho de uma solução satisfatória. Uma abordagem pertinente seria usar a computação,

tratando essa problemática como um problema de otimização combinatória, passível de ser resolvido através do uso de algoritmos.

Desta forma, o objetivo geral deste trabalho consiste em desenvolver um método de otimização de roteamento de veículos para a coleta de lixo de descarte irregular em área pública utilizando algoritmos de otimização. Buscando atingir o objetivo geral, tem-se alguns objetivos específicos tais como desenvolver uma metaheurística Greedy Randomized Adaptive Search Procedure (GRASP) e um Algoritmo Genético para o problema de roteamento de veículo, onde o GRASP ficará responsável pelo cálculo das rotas de menor caminho e o Genético será aplicado ao problema das p -medianas para realizar a divisão de um grafo em grupos, realizar os testes dos algoritmos e validar com dados reais da cidade de Mossoró-RN.

O trabalho encontra-se organizado da seguinte forma: o Capítulo 2 apresenta a fundamentação teórica sobre gestão de resíduos urbanos, otimização combinatória, problema do Roteamento de Veículos, problema das p -medianas, GRASP e Algoritmos Genéticos; o Capítulo 3 descreve os detalhes da implementação do modelo de otimização; o Capítulo 4 expõe os testes realizados e os resultados; o Capítulo 5 explana as considerações finais e os trabalhos futuros.

2. REFERENCIAL TEÓRICO

Nesta seção, apresenta-se a problemática do Resíduos Sólidos, como também, a explanação de conceitos que são fundamentais para desenvolvimento deste trabalho como: otimização combinatória, o problema do Roteamento de Veículos, o problema das p-medianas, heurísticas e metaheurísticas, GRASP e Algoritmos Genéticos.

2.1 Resíduos Sólidos

Trazendo para o contexto dos centros urbanos, os Resíduos Sólidos Urbanos (RSU) podem ser classificados quanto à origem e à periculosidade. De acordo com D'almeida et al. (2018), a origem pode ser classificada como: domiciliar, comercial, pública, serviços de saúde e hospitalar, portos, aeroportos, terminais rodoviários, ferroviários, industrial, agrícola e entulho. O lixo proveniente de domicílios, estabelecimentos comerciais (quantidades pequenas, geralmente até 50 quilos) e serviços públicos devem ser de responsabilidade da prefeitura enquanto que os outros ficam sob o cuidado dos seus geradores.

Segundo a Associação Brasileira de Normas Técnicas (ABNT), na Norma Brasileira (NBR) 10004/2004, Resíduos Sólidos (RS) são definidos sob o seguinte conceito:

“Resíduos nos estados sólido e semissólido, que resultam de atividades de origem industrial, doméstica, hospitalar, comercial, agrícola, de serviços e de varrição. Ficam incluídos nesta definição os lodos provenientes de sistemas de tratamento de água, aqueles gerados em equipamentos e instalações de controle de poluição, bem como determinados líquidos cujas particularidades tornem inviável o seu lançamento na rede pública de esgotos ou corpos de água, ou exijam para isso soluções técnica e economicamente inviáveis em face à melhor tecnologia disponível”. (ABNT, 2004, p1).

Conforme a periculosidade dos resíduos, a NBR 10004/2004 classifica da seguinte forma:

- Resíduos de Classe I (Perigosos): são os que possuem propriedades que apresentam risco à saúde pública e/ou ao meio ambiente. Propriedades como: inflamabilidade, corrosividade, reatividade, toxicidade e patogenicidade.

- Resíduos de Classe II A (Não inertes): são os que possuem alguma (s) das seguintes propriedades: combustibilidade, biodegradabilidade ou solubilidade e que não se enquadram em nenhuma das outras duas classes.
- Resíduos de Classe II B (Inertes): Não têm constituinte algum solubilizado em concentração superior ao padrão de potabilidade de águas.

O Gerenciamento de Resíduos Sólidos Urbanos na gestão de um município é fundamental. Essa administração ocorre através da gestão e do Gerenciamento Integrado de Resíduos Sólidos Urbanos (GIRSU). Segundo Junior (2003), neste contexto, gestão e gerenciamento assumem conotações diferentes. A gestão é a parte responsável por definir ações e procedimentos e nível estratégico enquanto que o gerenciamento está focado na operação do sistema de limpeza urbana.

Conforme a Lei No. 12305 (BRASIL, 2010), o gerenciamento integrado de resíduos sólidos urbanos trata-se de um conjunto de ações exercidas nas etapas de coleta, transporte, transbordo, tratamento e destinação final adequada dos resíduos sólidos. Ainda de acordo com essa lei, a gestão integrada consiste em ações objetivando soluções para os RS, levando em consideração dimensões políticas, econômicas, ambientais, culturais e sociais daquele município.

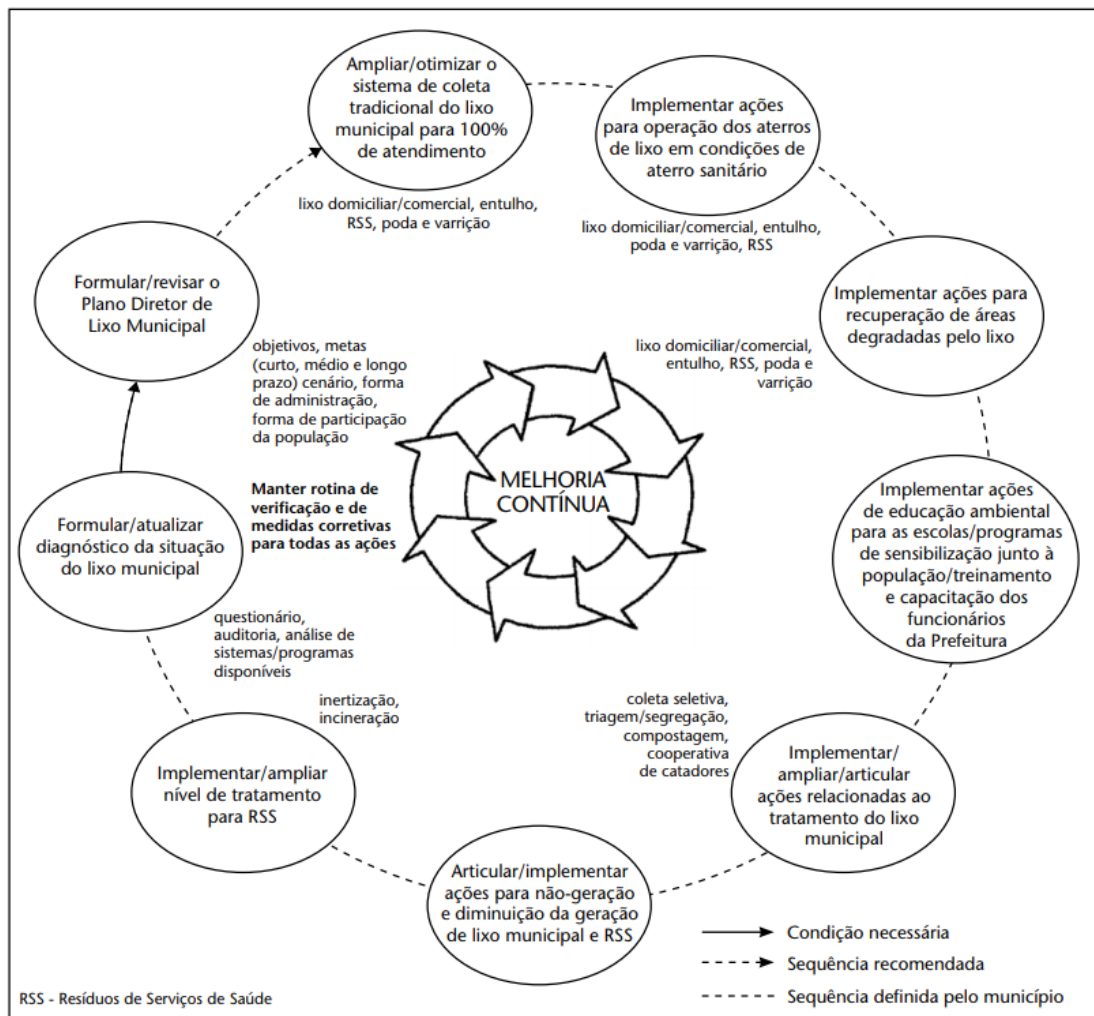
De acordo com D'almeida et al. (2018), gerenciar o lixo de forma integrada significa conseguir conceber um modelo que leve em consideração a quantidade e o tipo dos resíduos gerados pela população que mora naquela área, para desta forma garantir que o destino final seja ambientalmente correto, bem como o uso de tecnologias que sejam compatíveis com a realidade do local.

Além disso, também é importante que sempre haja a consciência por parte dos gestores de que todas as etapas são interligadas, desta forma, uma coleta mal planejada, acarretará em um transporte mal organizado, que gerará reclamações e assim surgirão falhas em todas as etapas subsequentes.

Nas etapas integradas, o GIRSU busca prevenir, ou ao menos reduzir a geração de RSUs. Segundo Junior (2003) a redução na fonte, o reaproveitamento, o tratamento e o destino final são aspectos que devem ser hierarquizados levando em

consideração as características que definem aquele ambiente. Também é pertinente a criação de metas, principalmente em lugares que a situação já se encontra em estágios mais graves, para que a análise dos resultados obtidos se torne mais fácil. A Figura 1 apresenta as etapas a serem seguidas.

Figura 1: Etapas integradas de um GIRSU.



Fonte: D'ALMEIDA et al. (2018).

É importante destacar que existem etapas que são condições necessárias para outras, que existem sequências de etapas recomendadas na literatura e também sequências que serão definidas dependendo da situação de cada município. As primeiras etapas devem estar focadas em recuperar o ambiente da sua atual situação.

Inicialmente se realizam ações para manutenção de aterros sanitários e recuperação de áreas danificadas pelos resíduos, bem como ações educacionais objetivando sensibilizar a população. Com tudo isso realizado, é hora de partir para ações direcionadas à manutenção dos Resíduos de Serviços de Saúde (RSS), que

apresentam propriedades bem perigosas, para que só depois possa ser diagnosticada a situação do município e aplicado um modelo para otimizar a coleta do lixo.

2.2 Otimização Combinatória

A otimização combinatória é uma área que busca por soluções que consigam realizar o melhor uso dos recursos envolvidos nos problemas estudados. As pesquisas desta área consistem em minimizar ou maximizar uma função aplicada a um domínio finito, de modo que a solução obtida otimize os objetivos de interesse. Alguns problemas clássicos incluem otimizar o tempo para ações e operações, definir as melhores rotas para transporte ou coleta de materiais, aumentar lucros, diminuir gastos, etc. (MIYAZAWA F. K., SOUZA C. C., 2015).

De acordo com Yang et al. (2013), a otimização tem um papel fundamental na busca por soluções aplicáveis aos problemas da vida real, em áreas como economia, ciência da gestão, administração, medicina, inteligência artificial, dentre outras. Yang et al. (2013) também afirmam que técnicas de otimização são desenvolvidas por matemáticos desde os primeiros anos da matemática, mas que a área só teve grande progresso com o surgimento dos computadores que permitiram a implementação de técnicas em estruturas de larga escala.

Existem inúmeros problemas de otimização, cada um com características diferentes. Yang (2011) defende alguns parâmetros que podem ser usados para classificar esses problemas. Como exemplo temos o número de objetivos, podendo ser um problema com objetivo único, com apenas uma função para minimizar ou multiobjetivo, que é preciso se preocupar com mais de uma função simultaneamente. Outro parâmetro é a quantidade de restrições que devem ser respeitadas. Em um contexto do mundo real é comum que o problema tenha mais de uma restrição e também seja multiobjetivo.

Na literatura existem vários métodos de resolução para problemas de otimização combinatória. Chaves (2009) afirma que uma forma de resolver esses problemas é testar as possibilidades possíveis e manter a melhor solução obtida, porém, quando se trata de problemas reais, esse método se torna impraticável, pois o número de soluções possíveis cresce exponencialmente em função do problema,

logo, técnicas mais apuradas são necessárias. Um exemplo de problemas reais é o roteamento de veículos.

2.3 Problemas de Roteamento de Veículos

O problema de Roteamento de Veículos (*Vehicle Routing Problem*, ou VRP) examina como atender a um número de clientes utilizando um conjunto de veículos, localizados em um ou mais depósitos, que se locomovem através de uma malha viária em um dado espaço de tempo. A solução final do VRP consiste em um conjunto de rotas em que todos os clientes são atendidos, as constantes globais são satisfeitas e o custo de transporte geral é otimizado. Esse problema pode ser aplicado em várias situações diferentes como: coleta de resíduos sólidos, roteamento de ônibus escolares, roteamento de vendedores, entre outros. (VIGO; TOTH, 2002).

O problema é dado por um conjunto finito C de clientes que residem em N localizações diferentes. Cada par i, j de localização possui uma distância D_{ij} . Tal distância pode ser simétrica, ou seja $D_{ij} = D_{ji}$. Também existe o conjunto finito V que representa a frota de veículos, e eles têm uma capacidade K que pode ser igual para todos ou não. Caso K seja o mesmo em todos os veículos, teremos um problema com veículos homogêneos. Caso contrário, serão heterogêneos.

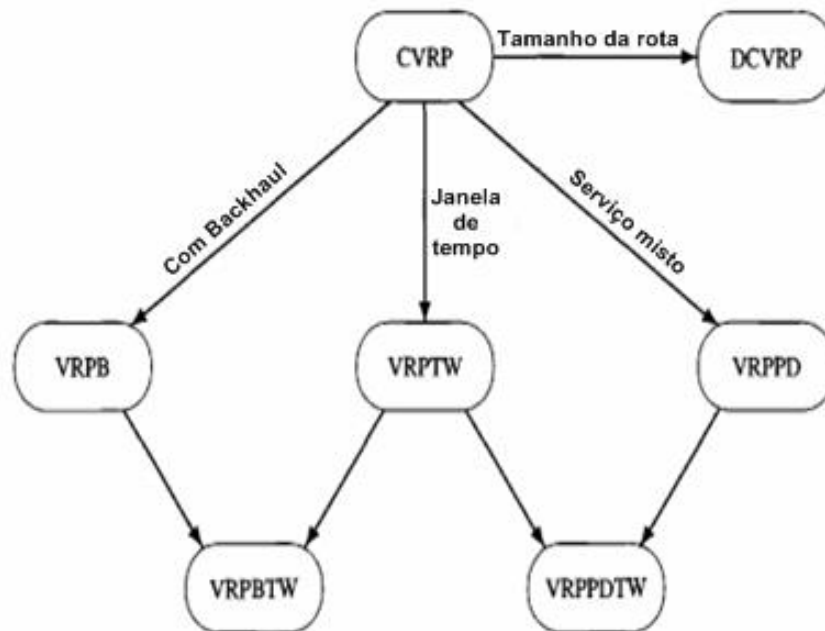
Os problemas deste tipo são representados através de grafos, que são estruturas compostas por arestas, que correspondem as estradas, e vértices, que representam localização dos clientes. De acordo com Vigo e Toth (2002), as variações básicas do problema são:

- **Roteamento de Veículos Capacitado (CVRP):** Os veículos são homogêneos, e possuem uma capacidade total. O objetivo de encontrar um conjunto de rotas tais que cada cliente seja visitado por exatamente um veículo e a soma das demandas dos vértices de uma rota não ultrapasse a capacidade do veículo escolhido.
- **Roteamento de Veículos com Restrição de Distância (DCVRP):** É bem semelhante ao CVRP, entretanto as restrições de capacidade dos veículos são substituídas por restrições de distâncias das rotas. Na prática, isso significa que o tamanho de cada circuito não pode exceder um valor estipulado como tamanho máximo.

- **Roteamento de Veículos com Janela de Tempo (VRPTW):** É uma extensão do CVRP, em que é adicionado um intervalo de tempo chamado de janela de tempo. Além de todas as restrições do CVRP, nesta variação os veículos devem iniciar dentro da janela de tempo estabelecida e permanecer em cada cliente uma quantidade de tempo representando o atendimento. Caso algum veículo chegue antes do que estava previsto, ele fica aguardando até que se inicie a janela de tempo.
- **Roteamento de Veículos com *Backhauls* (VRPB):** Também é uma extensão do CVRP que além das restrições de capacidade, é dividido em dois subconjuntos. O primeiro subconjunto contém uma N quantidade de clientes *Linehaul*, que possuem uma demanda de recebimento de algum produto. O outro subconjunto é formado por M clientes *Backhaul* que possuem uma quantidade de produtos que precisam ser coletados. Uma rota obrigatoriamente deve ter clientes *Linehaul*, podendo ou não ter clientes *Backhaul*. A principal restrição dessa variação acontece quando uma rota passa pelos dois subconjuntos. Desta forma, é preciso obedecer uma ordem de precedência e primeiro deve ser realizada a entrega antes da coleta.
- **Roteamento de Veículos com Retiradas e Entregas (VRPPD):** Essa versão se trata de outra extensão do CVRP e nela cada cliente possui duas quantidades, representando a demanda de mercadorias que devem ser entregues e coletadas, nessa ordem. Para cada vértice i , O_i representa o vértice que é a origem da demanda de entrega e D_i representa o vértice que é o destino da retirada.

Na literatura podem ser encontradas diversas variações do problema. Existem também as variações que são junções de duas ou mais básicas, exibidas na Figura 2, e outras que foram surgindo com o tempo. De acordo com Golden, Raghavan e Wasil (2008), nos últimos anos surgiram outras variantes devido a inúmeras evoluções tecnológicas, como: *Global Positioning System* (GPS), identificação por radiofrequência e computação paralela. É importante ressaltar que tais versões possuem um tempo computacional não-polinomial.

Figura 2: Variações básicas do VRP.



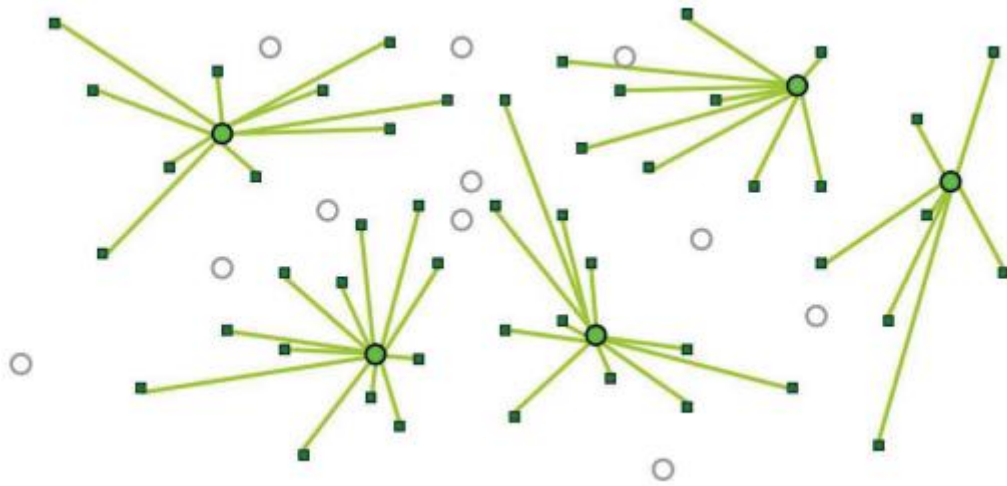
Fonte: Adaptado de VIGO; TOTH (2002).

2.4 Problemas das p-medianas

O problema das p-medianas sem custo (PMNC) é um problema de localização de facilidades que busca definir um número p de nós (denominados de medianas) em uma rede, objetivando minimizar a distância total necessária para se atender todas as demandas. Isler, Bonassa e Cunha (2012) argumentam que existe uma particularização desse problema conhecida como o problema das medianas capacitado (PMC), que adiciona restrições nos candidatos, de modo que cada ponto de demanda seja alocado sem violar a capacidade de cada mediana.

Assim, dado um grafo $G = (V, E)$, que possui vários pontos de demanda espalhados, se busca encontrar um subconjunto de medianas $V_p \in V$ tal que a soma das distâncias entre as demandas e sua mediana mais próxima seja o mínimo possível. De acordo com Pereira (2005) os dados relevantes desse problema são: o número finito, tanto de pontos de demanda quanto de candidatos, e a distância entre os pontos de demanda e o número p de facilidades a serem encontradas. A solução de um problema deste tipo é apresentada na Figura 3.

Figura 3: Uma resolução do problema das p-medianas.



Fonte: Adaptado de SOARES (2009).

Segundo Goldengorin, Krushinsky e Pardalos (2013), este problema é não-polinomial e pesquisado amplamente na literatura, sendo aplicado a várias situações diferentes, tais como: análise de *cluster*, psicologia quantitativa, *marketing*, indústria de telecomunicações, roteamento de veículos, desenvolvimento de topologias e de redes de computadores, etc.

2.5 Heurísticas e Metaheurísticas

Esses métodos vêm se tornando cada dia mais conhecidos, apesar disso, é relativamente comum na literatura encontrar pesquisadores que usam os termos “metaheurística” e “heurística” de forma intercambiável. Entretanto, há diferenças.

De acordo com Yang (2011), heurísticas são uma solução estratégica que através de tentativa e erro produzem um resultado que é aceitável em um intervalo de tempo prático. Ao testar todas as possibilidades, a solução ótima é encontrada, porém os recursos são limitados. Desta forma, o objetivo das heurísticas é garantir um resultado bom em um tempo realizável, sem se preocupar que aquele resultado seja o melhor possível.

As heurísticas são usadas, geralmente, quando se trabalha com problemas que são não-polinomiais em instâncias reais, e a obtenção de um resultado satisfatório demanda uma grande quantidade de tempo, tendendo a aumentar exponencialmente

em função do tamanho da entrada. Para a este tipo de resolução é comum recorrer aos algoritmos exatos ou heurísticas.

Ainda de acordo com Yang (2011), a primeira pessoa a utilizar uma heurística com o auxílio de um computador foi Alan Turing durante a Segunda Guerra Mundial. Ele, juntamente com Gordon Welchman, desenvolveu uma máquina de criptoanálise que utilizava um algoritmo heurístico para quebrar a criptografia Alemã. Outro avanço nesta área foi nas décadas de 1960 e 1970 quando pesquisadores da Universidade de Michigan desenvolveram os primeiros algoritmos genéticos que se tem conhecimento.

Dois conceitos importantes dentro desse tema são o ótimo local e o ótimo global. Colin (2007) propõe uma boa analogia sobre como funcionaria um processo heurístico para localizar empiricamente o lugar mais alto da terra. Primeiro, partir-se-ia de um ponto qualquer da superfície da Terra em busca de montanhas mais altas. Seguindo o processo, várias montanhas seriam encontradas e suas alturas comparadas. No decorrer da descoberta, o ponto mais alto iria progressivamente aumentando com as novas descobertas (os chamados ótimos locais), até o fim das buscas. Assim, a solução retornada é a melhor encontrada até a parada. Em nenhum momento se garante que o resultado encontrado seria o melhor possível (ótimo global) e sim que ele seria um bom resultado.

As metaheurísticas podem ser entendidas como estruturas de algoritmos gerais, sendo assim adaptáveis a vários problemas diferentes. Yang et al. (2013) afirmam que as metaheurísticas fazem um balanço entre a busca local e a exploração global, e a variedade de soluções é realizada a partir de aleatoriedade, e buscas em escala global. Assim como as heurísticas, estes métodos também não garantem resultados ótimos e sim um tempo realizável.

Blum e Roli (2003) afirmam que os dois componentes principais de uma metaheurística são a diversificação e a intensificação, enquanto Yang (2011) defende que a seleção também é uma parte importante. Dessa forma, a diversificação faz com que hajam soluções das mais variadas possíveis, além de ajudar a fugir de regiões de ótimos locais, a intensificação é responsável pela busca em uma região local, explorando as possibilidades e a seleção garante que apenas as melhores soluções, ou aquelas que tem uma chance alta de convergir para o ótimo, sejam mantidas.

Ainda de acordo com Yang (2011) uma das formas de classificar as metaheurísticas é dividi-las baseadas em população e conforme a trajetória. Um exemplo desse primeiro tipo são os Algoritmos Genéticos. Nessa técnica se usa de uma população de cromossomos que representam soluções. Já como exemplo de trajetória temos o Recozimento Simulado (Simulated Annealing) que faz uso de uma solução inicial que vai percorrendo etapas, de forma que os membros com uma solução melhor sempre serão aceitos enquanto que uma solução não tão satisfatória também poderá ser aceita, mas de acordo com uma probabilidade. Assim, os passos formam uma trajetória.

2.6 GRASP

A metaheurística GRASP, inicialmente descrita por Feo e Resende (1989), que em português significa processo de busca adaptativa, randômica e gulosa é bastante utilizada na literatura na resolução de problemas de otimização. É um método que recebe como parâmetro um α e uma quantidade de iterações divide-se em duas fases. A primeira é dita a fase de construção da solução, que consiste em criar uma solução elemento a elemento enquanto a outra é a fase de busca local, que realiza uma varredura na vizinhança, a solução atual é melhorada.

É uma técnica que possui algumas variações, podendo ter suas fases implementadas de forma que melhor se adapte ao problema e assim equilibrando a parte gulosa com a aleatória. O GRASP na sua versão básica, apresenta apenas dois parâmetros, logo se economiza tempo no ajuste destes. A versão a seguir do pseudocódigo deste algoritmo foi adaptada de Resende e Feo (2016).

```

Função GRASP(maxIterações, alfa)
1. melhorCusto =  $\infty$ ;
2. Para  $k = 1, \dots, \text{maxIterações}$  Faça
3.      $S = \text{SoluçãoGulosaAleatória}(\text{alfa});$ 
4.     Se  $S$  não é realizável Então
5.          $S = \text{RepararSolução}(S);$ 
6.     Fim;
7.      $S = \text{BuscaLocal}(S);$ 
8.     Se  $\text{calcularCusto}(S) < \text{melhorCusto}$  Então
9.          $\text{melhorSolução} = S;$ 

```

```

10.         melhorCusto = calcularCusto(S);
11.     Fim;
12. Fim;
13. Retorne S*;

```

2.6.1 Construção da Solução

Nessa fase é realizada a construção da solução, que pode seguir um critério guloso, aleatório ou um balanço entre esses dois. Inicialmente é adicionado na solução o primeiro vértice ou, dependendo da implementação, um vértice aleatório. A partir disso, é criada uma Lista de Candidatos (LC) a serem adicionados e tal lista é ordenada de forma gulosa, isto é, são ordenados em função da distância entre eles e o vértice atual. Forma-se então uma Lista Restrita de Candidatos (LRC).

O parâmetro α receberá valores entre 0 e 1. De acordo com Resende e Ribeiro (2016), o α define o quão gulosa ou aleatória a construção será. De forma que quando $\alpha = 0$ selecionamos apenas o primeiro elemento da LRC, assim teremos uma construção puramente gulosa, enquanto que com $\alpha = 1$ selecionaremos elementos de forma aleatória presente na LRC e qualquer valor entre esses a construção é balanceada entre gulosa e randômica.

2.6.2 Busca Local

Essa segunda etapa visa melhorar a solução já criada através da realização de uma busca local na sua vizinhança. A maneira que essa busca é implementada pode variar. Como a solução construída serve de entrada para esta etapa, Resende e Ribeiro (2016) afirmam que é imprescindível sempre checar se uma solução é realizável. Caso ela não seja, deve passar por um processo de reparação.

Algumas técnicas de otimização que podem ser utilizadas são: Subida da Montanha (Hill Climbing), Recozimento Simulado, Busca em Vizinhança Variável (VNS), entre outras. Através da troca entre vértices vizinhos, o GRASP objetiva sempre atingir uma melhora nas soluções.

2.7 Algoritmos Genéticos

Os Algoritmos Genéticos (AG) são métodos de otimização e busca baseados nos princípios da genética e da seleção natural. Um AG trabalha com uma população

de vários indivíduos contendo genes que representam as soluções e que evoluem sob algumas regras, objetivando maximizar a aptidão e dessa forma minimizar os custos. Tal método foi inicialmente proposto por Holland (1975), além de ser estudado e popularizado pelos estudantes da sua universidade.

Yang (2011) afirma que muitas metaheurísticas são inspiradas em processos que ocorrem na natureza e, no caso dos AGs, isso não é diferente. As espécies são capazes a se adaptar ao ambiente que vivem, na maioria das vezes desenvolvendo estruturas complexas que permitem que vivam em mais de um local. De acordo com Kramer (2017) estas são boas razões para adaptar princípios da evolução na resolução de problemas de otimização.

2.7.1 Estrutura de um Algoritmo Genético

Existem variantes desse método, entretanto segundo Kramer (2017), a versão básica de um AG possui três operadores: cruzamento, mutação e evolução. Cada operador pode ser implementado de formas diferentes. Outra questão pertinente é a de parâmetros, que também podem ser implementados de modos distintos, sendo possível uma adaptação ao problema estudado. Conforme Yang (2011), os passos básicos dessa abordagem são:

- Definir a função de otimização;
- Inicializar a população de indivíduos;
- Avaliar a aptidão de cada membro da população;
- Modificar a população através de cruzamentos, mutações, etc.;
- Evoluir a população, mantendo apenas os melhores indivíduos.

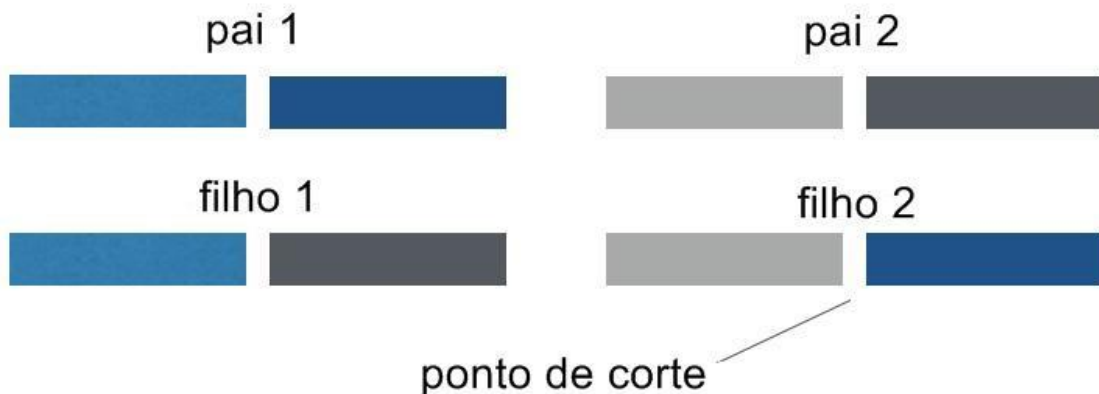
Primeiramente a população deve ter todos os seus indivíduos inicializados. Depois, alguns deles serão escolhidos para participarem de um cruzamento, gerando novos filhos e membros para a população. Eles também poderão passar por uma mutação para que seja adicionada um pouco de aleatoriedade nas soluções, e é realizado o cálculo da aptidão de cada indivíduo com base na função de otimização. No fim de cada iteração, ocorre a seleção, onde apenas os melhores indivíduos continuarão.

2.7.2 Cruzamento

Esse operador permite que dois indivíduos tenham sua solução combinada, gerando uma nova que deve ser atribuída a indivíduos filhos. Kramer (2017) afirma

que um meio comum de representar o cruzamento em um AG é através do cruzamento de n -pontos. Nessa abordagem, são escolhidos n pontos onde duas soluções serão divididas. Através da junção dos pedaços, novas soluções são formadas. Tal processo é ilustrado na Figura 4. Essa solução se baseia na premissa que a união de partes de boa qualidade produzirá filhos também com boa qualidade.

Figura 4: Representação gráfica da implementação de um cruzamento.



Fonte: Adaptado de KRAMER (2017).

Para o caso de problemas específicos, o cruzamento pode resultar em uma solução não realizável. Neste caso, é necessário haver algum procedimento de reparação. Também é importante que o cruzamento funcione a partir de um parâmetro de taxa, caso contrário, a cada iteração realizada, a população está propensa a riscos de se tornar homogênea, perdendo o fator de aleatoriedade.

2.7.2 Mutação

A mutação é um operador cuja função principal é realizar alterações aleatórias em algumas soluções, de forma que ajude a sair de ótimos locais. De acordo com Yang (2011), essa operação pode ser alcançada ao selecionar aleatoriamente genes de uma solução e alterar os valores. Em um problema baseado em definir caminhos, uma forma comum de implementar é através da escolha de dois vértices, sendo que um é trocado de posição com o outro.

Esse operador é executado de acordo com o parâmetro de taxa de mutação, onde normalmente usa valores baixos. Uma alternativa é não permitir que os melhores indivíduos da população sofram mutação, pois por mais que eles possam melhorar, existe uma chance maior de acontecer o contrário. Dessa forma é garantido que as soluções de maiores qualidades permaneçam sem mudança.

2.7.3 Seleção

Para que o cruzamento ocorra corretamente é preciso uma maneira de selecionar apenas os indivíduos com uma alta aptidão para os papéis de pais. A seleção é uma etapa importante dentro dos AGs. Kramer (2017) argumenta que métodos elitistas produzem melhores resultados. Existem vários algoritmos de seleção na literatura. Entre esses algoritmos se destaca o da Roleta, também conhecido como seleção proporcional a aptidão. Em tal abordagem a probabilidade de um indivíduo ser selecionado está diretamente relacionada com o seu valor de aptidão.

Nos AGs também é necessária uma condição de seleção que determine quais indivíduos irão sobreviver a cada geração da população. Esse operador pode ser implementado em conformidade com princípio Darwiniano de sobrevivência dos mais fortes, dessa forma a cada criação de uma nova população, são mantidos apenas os membros com maior aptidão, enquanto que os classificados como piores são substituídos, dando lugar a novas soluções.

2.8 Trabalhos Relacionados

Burkhal, Larsen e Ropke (2012) abordam a problemática da coleta de lixo em uma cidade como um problema de roteamento de veículos com janela de tempo, de forma que sejam calculadas rotas para que os caminhões de lixo consigam coletar os resíduos, respeitando restrições de tempo dos clientes e garantindo intervalos de descanso para os motoristas. Eles propõem uma metaheurística conhecida como Adaptive Large Neighborhood Search (ALNS), que traduzido significa Busca Adaptativa em Vizinhança de Grande Porte.

Outro trabalho relevante é o de Benjamin e Beasley (2010) que também considera o tempo de descanso para os motoristas, além de garantir múltiplos pontos de despejo. Os autores resolvem o problema através da aplicação de metaheurísticas, como a Busca Tabu e a Busca em Vizinhança Variável.

Já o trabalho de Ramakrishnaiah, Shiju e Vathsala (2011) propõe um método de otimização da coleta de resíduos sólidos de um município através do uso de Sistemas de Informações Geográficas (SIG). O modelo desenvolvido para geração de

rotas leva em consideração alguns parâmetros, como: população, capacidade de geração de resíduos, malha viária, tipos das estradas e os veículos de coleta.

3. IMPLEMENTAÇÃO

Nesta seção, serão exibidos os aspectos relevantes utilizados para a implementação de um modelo de otimização para a coleta do lixo irregular em lugares públicos, abordando tal problemática como um Problema de Roteamento de Veículos. Para a resolução, são utilizadas duas metaheurísticas: um Algoritmo Genético aplicado ao problema das p -medianas e um GRASP resolvendo o caminho hamiltoniano dada uma matriz de adjacências. O objetivo é, a partir de uma lista de coordenadas que representam a localização dos resíduos e uma quantidade de veículos, determinar uma rota otimizada para cada veículo realizar a coleta. É possível ter acesso ao código-fonte do algoritmo acessando o link a seguir: <https://github.com/mahteusz/monografia>.

Na literatura existem algumas aplicações de AGs ao problema das p -medianas. É possível observar exemplos de tal aplicação nos trabalhos de Isler, Bonassa e Cunha (2012) e Alp e Erkut (2003), onde resultados satisfatórios foram obtidos. O principal motivo da escolha do AG é devido a existência de um conjunto de soluções que são viáveis e que mesmo após um cruzamento continuarão viáveis sem a necessidade de um procedimento de reparação.

Também já foram desenvolvidas muitas aplicações do GRASP a problemas de roteamento de veículos. Os trabalhos de Crainic et al. (2013) e Brito et al. (2009) apresentam tal metaheurística conseguindo gerar bons resultados quando aplicada a extensões desse problema.

Os algoritmos foram desenvolvidos utilizando a linguagem Python, manipulando a biblioteca *rand*, que fornece funções para geração de valores pseudo-aleatórios e fazendo uso também da biblioteca *requests*, que auxilia as requisições do protocolo *HyperText Transfer Protocol* (HTTP). Na parte que lida com as coordenadas geográficas, foi utilizada uma *Application Programming Interface* (API) chamada *Bing Maps API*, implementada pela Microsoft, que fornece vários serviços *Representational State Transfer* (REST), permitindo a realização de requisições sobre informações geográficas. Os motivos principais para a escolha dessa API sucederam pelo fato que além de todas as funcionalidades, há uma facilidade na obtenção de uma chave gratuita para fins educacionais.

3.1 Estrutura Geral

As partes que compõem o algoritmo foram implementadas de forma independente, onde cada uma receberá o resultado da sua anterior como entrada e proverá a entrada da parte seguinte. Tal abordagem é importante pois caso haja alguma falha ou necessidade de mudança em alguma etapa, as outras não serão atingidas. O modelo implementado consiste em quatro partes distintas, além do programa principal, que é responsável por realizar a intermediação entre todos os seguintes componentes:

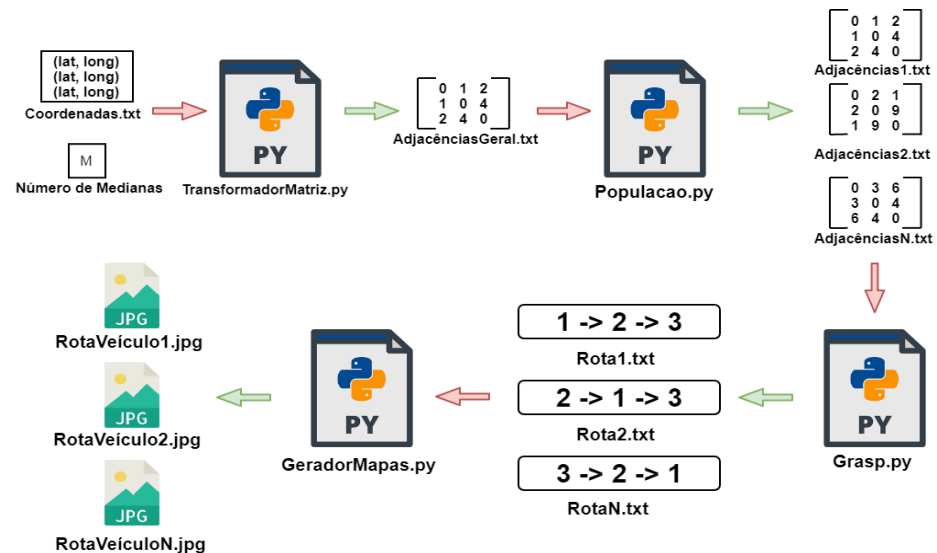
- Classe GRASP: é onde está implementado todo o código dessa metaheurística, com a construção de uma solução, a busca local e laço principal.
- Classe Indivíduo: implementa a representação e todos os métodos de um Indivíduo que compõe a população do Algoritmo Genético.
- Classe População: é responsável pela população do AG, que é composta por vários Indivíduos e codifica todas as operações que um AG deve ter.
- Classe TransformadorMatriz: responsável por transformar uma lista de coordenadas geográficas em uma matriz de adjacências.
- Classe GeradorMapa: classe que a partir dos resultados conseguidos, gera uma imagem mostrando o caminho com os pontos.

Os algoritmos devem ser executados em uma ordem definida para que haja um resultado satisfatório, pois, a saída de uma classe serve como entrada para outra. Isto posto, a sequência de passos que deve ser seguida, ilustrada na Figura 5, é a seguinte:

1. É recebida a lista de coordenadas geográficas e um número m de veículos como entrada;
2. As coordenadas são transformadas em uma matriz de distâncias;
3. A matriz de distâncias é dividida em m grupos, sendo uma matriz para cada grupo, que representam a rota que deve ser seguida por cada veículo;

4. Para cada matriz da etapa anterior, é calculado um caminho otimizado;
5. A partir dos caminhos resultantes, são geradas imagens que representam as rotas que devem ser seguidas.

Figura 5 - As etapas do algoritmo desenvolvido.



Fonte: Autoria própria.

3.1.1 Transformador de Matrizes.

Muitos problemas de otimização combinatória são tratados como grafos com pesos e as abordagens deste trabalho também atendem aos preceitos da literatura. Inserir, manualmente, a posição de vários lugares do mundo real com a distância entre eles é uma tarefa trabalhosa e propensa a erros. Então, fez-se necessário a implementação dessa classe para que a partir de coordenadas geográficas fosse possível construir automaticamente as adjacências de um grafo, fazendo com que os algoritmos tenham um funcionamento correto.

Dentre os serviços *REST* fornecidos pela API da Microsoft, essa classe utiliza algumas funcionalidades da *Distance Matrix API*, que é utilizada para resolver esse problema, possibilitando calcular distâncias em rotas de pedestres ou de veículos.

As requisições são realizadas através do método *GET* do protocolo HTTP, onde é necessário a passagem de uma *Uniform Resource Locator* (URL) com parâmetros. Os principais parâmetros utilizados são:

- *Origins*: esse parâmetro recebe uma lista de pares, com a latitude e longitude de cada coordenada de origem.
- *Destinations*: parâmetro bem semelhante ao *origins*, a diferença é que aqui é onde ficará a lista com as coordenadas de destino.
- *travelMode*: é definido o modo de transporte que será utilizado quando as distâncias estiverem sendo calculadas. Como esse trabalho aborda roteamento de veículos, o valor utilizado foi o de *driving*, do inglês dirigindo.

Além desses parâmetros supracitados, ainda existem outros opcionais, que servem para definir a unidade de distância (a unidade padrão é em quilômetros), a unidade de tempo, etc. As respostas são retornadas no formato de *JavaScript Object Notation* (JSON) com alguns campos.

A classe realiza o processamento deste JSON, mantendo apenas as informações que são relevantes, neste caso as distâncias. A partir disso, a matriz é organizada e salva, já que será consumida como entrada para o próximo passo da otimização.

3.1.2 Algoritmo Genético

Na codificação do AG a primeira etapa foi decidir como uma solução para o problema das p -medianas seria representada através de um indivíduo. A forma adotada foi, a partir de um número m de medianas definido por parâmetro, um cromossomo é preenchido com um número de valores igual ao número de localizações existentes. Cada gene é preenchido com números que representam as medianas enquanto que os índices do cromossomo representam os pontos de demanda. Para facilitar na visualização das medianas, os índices correspondentes a elas serão preenchidos com o valor de -1. A Figura 6 ilustra como se atribui uma possível representação de um indivíduo com medianas nos vértices 1 e 4, onde os pontos 0, 5 e 7 pertencem a mediana 1 enquanto que 2, 3 e 6 pertencem a mediana 4.

Figura 6 - Representação de um indivíduo.

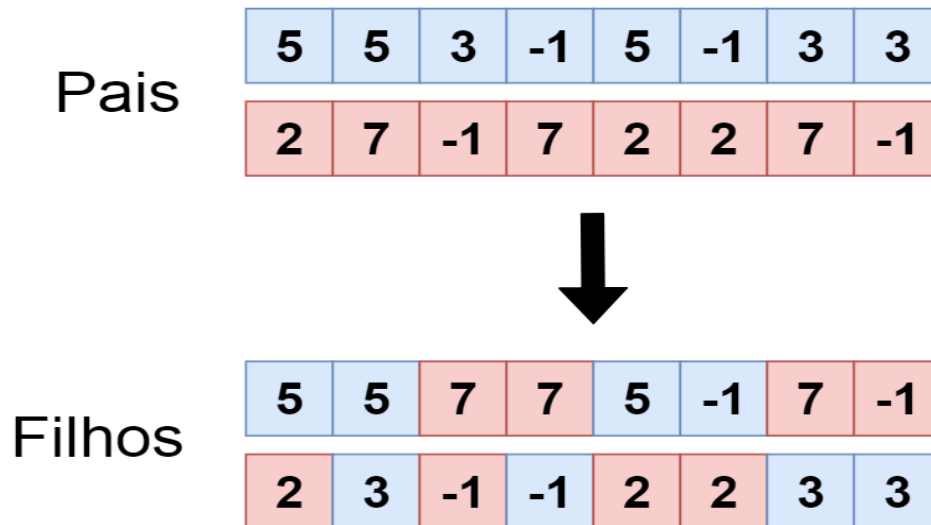
0	1	2	3	4	5	6	7
1	-1	4	4	-1	1	4	1

Fonte: Autoria própria.

Uma fase importante no desenvolvimento de um AG consiste em identificar qual o método de seleção. Nesta implementação foi escolhido o algoritmo de seleção por roleta, onde os indivíduos da população são ordenados de acordo com a função objetivo, de forma enquanto maior for a aptidão de determinado indivíduo, maior também será a chance deste ser escolhido. Essa abordagem é bem popular, pois consegue escolher pais com uma boa solução, mas sem abandonar totalmente a chance de escolher algum membro menos adaptado. Outra particularidade adotada nesta etapa é a de salvar uma porcentagem da população, objetivando preservar uma determinada parcela de indivíduos de serem escolhidos como candidatos a filhos para um cruzamento.

Depois que são escolhidos os membros de um cruzamento, dois indivíduos pais e dois para serem substituídos pelos filhos, a operação é realizada. Por se tratar de um problema com algumas singularidades, o cruzamento implementado se baseia em juntar as medianas dos pais e distribuí-las de forma aleatória entre os filhos, entretanto conservando as posições. Desta forma é possível que um filho herde metade das medianas de cada pai, ou uma quantidade maior de algum deles. A verificação é sempre realizada se o número final de facilidades é igual ao parâmetro estabelecido, com o intuito de que todo indivíduo que resulte de um cruzamento seja válido. A abordagem escolhida para um cruzamento é representada na Figura 7.

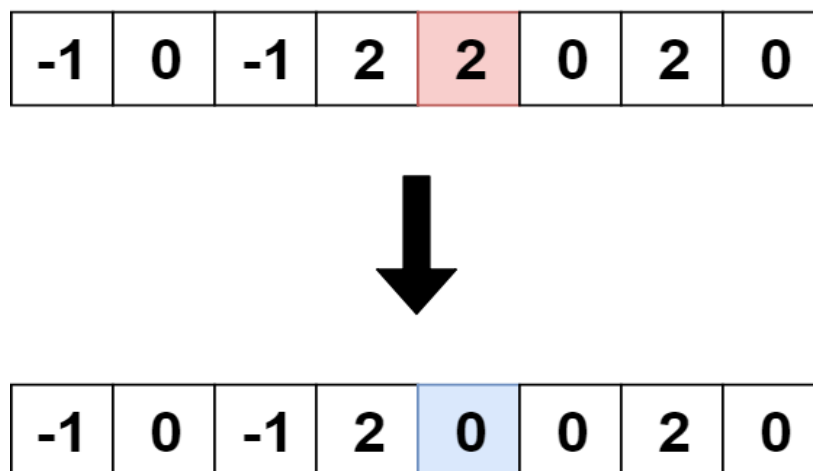
Figura 7 - Resultado de um cruzamento.



Fonte: Autoria própria.

A mutação ocorre primeiramente selecionando aleatoriamente um membro da população além de um gene dele e um ponto de facilidade. Depois disso é realizada a troca entre a facilidade que estava alocada na posição sorteada com a facilidade que também foi escolhida randomicamente. Uma alteração deste modo pode melhorar uma solução como também complicar, isto posto, existe um parâmetro que permite preservar os melhores indivíduos de sofrerem mutação. É possível visualizar essa operação na Figura 8.

Figura 8 - Representação de uma mutação.



Fonte: Autoria própria.

No laço principal, onde ocorre a evolução dos indivíduos, após a realização das etapas de ordenação, cruzamento e mutação, é realizada um operador chamado de

“repopulação”. Esse operador faz com que apenas os membros mais aptos sobrevivam, descartando uma grande parcela da população preenchendo as vagas com novos indivíduos com novas soluções, prontos para passarem por todos os modificadores e evoluírem. Tal repopulação também foi parametrizada, de forma que é possível optar por quantos por cento da população será substituída.

Antes de iniciar o algoritmo é preciso definir alguns parâmetros, fazendo com que seja possível adaptá-lo à diferentes instâncias. Além dos já supracitados, também temos: Taxa de cruzamento, taxa de mutação, número de gerações, tamanho da população, número de medianas. O resultado final são matrizes de adjacência, uma para cada mediana.

3.1.3 GRASP

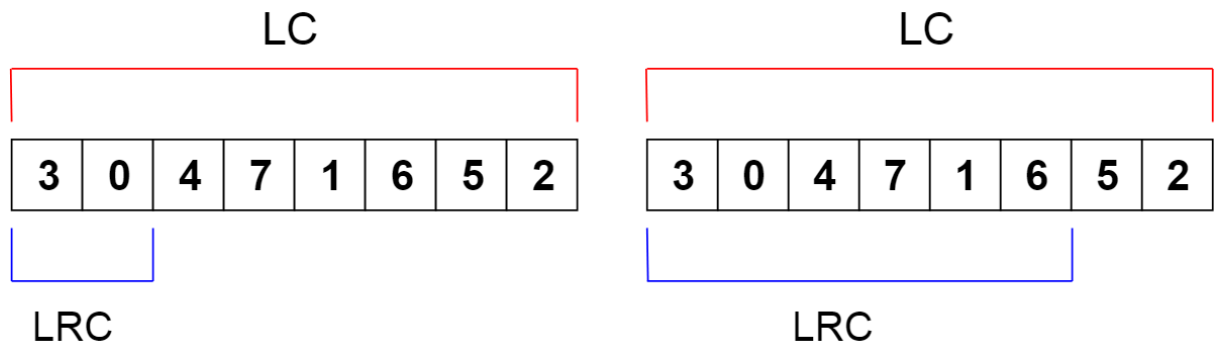
O algoritmo recebe como entrada um número de iterações, um alfa e uma matriz de adjacências entre as localizações geográficas e consiste em três partes principais: a construção de uma solução (que é representada através de um vetor com índices representando os pontos), a busca local e o laço principal. O objetivo é minimizar o custo, que é calculado a partir da soma das distâncias entre os pontos. A partir desses dados, um vértice inicial aleatório é sorteado para ser o primeiro da solução. Depois desse procedimento, os outros vértices serão selecionados em um laço, até a que a solução disponha de um caminho entre todas as localizações.

Nesse laço primeiramente é criada uma Lista de Candidatos, que será ordenada de forma gulosa com relação a localização atual. Utilizando o parâmetro alfa e a LC, é criada a Lista Restrita de Candidatos, que é calculada com o alfa multiplicado pelo tamanho da LC, que resulta em quantos candidatos da LC também irão para LRC. Por fim, é escolhido de forma aleatória um vértice da LRC para ser adicionada a solução.

Essa forma de organizar a LRC permite que o algoritmo consiga realizar escolhas gulosas (quando o valor de alfa é próximo de zero), aleatórias (com alfa próximo de um) ou um meio termo entre estas, permitindo que várias possibilidades da construção de um caminho possam ser testadas. Após a implementação de uma solução, o algoritmo realizará a busca local, realizando mudanças nos vértices,

objetivando diminuir o custo. A Figura 9 mostra o resultado da LRC com um valor de alfa de 0.2 e o como ficaria com um alfa definido como 0.8.

Figura 9 - Influência do valor de alfa.



Fonte: Autoria própria.

A maneira escolhida para a realização da busca local é baseada na heurística Escalada de Montanha, onde são realizadas alterações de estados e comparações de resultados. Para cada vértice é realizado os seguintes passos:

1. É feita a troca entre a localização atual e uma localização j ;
2. A nova solução obtida tem seu custo calculado;
3. j é incrementado;

Por fim, é mantida apenas a solução com a troca que apresentou o menor custo, logo é bem provável que haja uma melhora na solução. A condição de parada deste método é quando uma iteração é realizada e não há melhora, sendo retornada à solução gerada. Tanto a fase da construção de uma solução quanto a de busca local acontecem em um laço. Após esse laço principal, o resultado com a melhor rota servirá como entrada da próxima etapa da otimização.

3.1.4 Gerador de Mapas

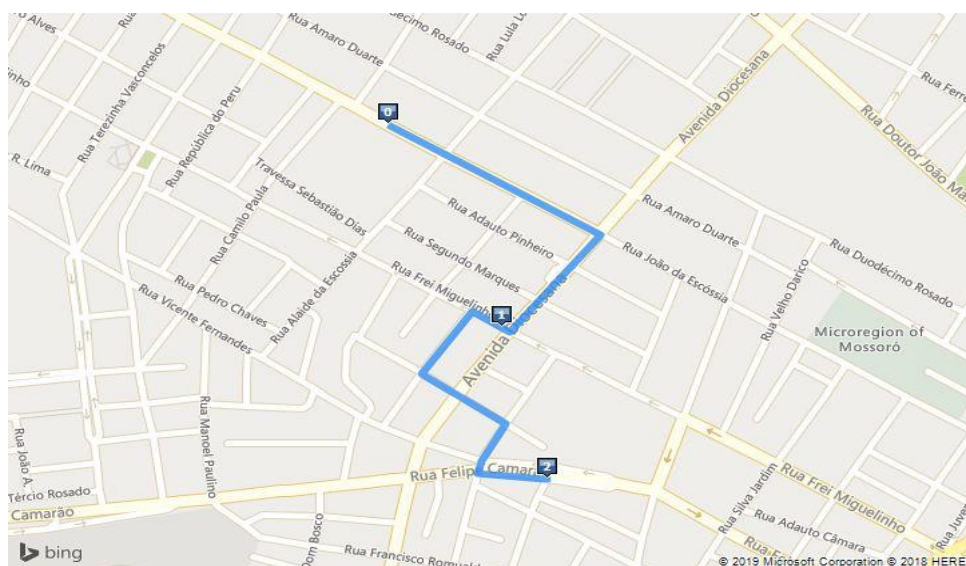
Depois que todas as rotas foram calculadas, seria pertinente um meio de visualizar exatamente qual a ordem que será seguida e por quais ruas especificamente. A implementação dessa classe veio para completar essa lacuna e permitir uma melhor representação dos resultados obtidos, detalhando através de uma imagem gerada todo o caminho que deve ser seguido em um mapa do local.

Na codificação também foi utilizada outra parte da *Bing Maps* API, que permite a geração de alguns tipos de mapas através de parâmetros. O modelo escolhido foi um mapa que possibilite a exibição de uma rota marcando os pontos e o caminho que seja centralizado automaticamente de acordo com as marcações. A requisição também foi feita através de um *GET* e, além da *key*, conta com os seguintes parâmetros principais:

- *imagerySet*: esse parâmetro está relacionado com o tipo de imagem que será obtido. Entre os valores possíveis, a implementação optou pela escolha do *Road*, que se trata de uma versão onde o mapa é representado através das estradas ou ruas, possibilitando uma melhor visualização da cidade.
- *Waypoints.n*: define-se onde são passadas todos os *n* pares de coordenadas,
- *mapSize*: parâmetro que definirá qual será o tamanho da imagem retornada. Neste trabalho optou-se por definir uma resolução de 720x480.

Também é possível alterar outros parâmetros opcionais, tais como o formato da imagem, formato da resposta, se é considerar apenas rodovias, dentre outros. A Figura 10 ilustra o resultado de uma requisição *GET*, com os pontos (-5.185202, -37.356895), (-5.188162, -37.354578), (-5.191068, -37.353966) e o *imagerySet* definido como *Road*.

Figura 10 - Imagem gerada com resultado de uma rota.



Fonte: Autoria própria.

3.1.5 Programa Principal

O programa principal é o responsável por instanciar todas as classes descritas acima, além de controlar e direcionar as saídas de cada etapa. Assim, são definidos todos os parâmetros, sendo a lista de coordenadas e o número de medianas como requeridos, e também um *log* que mostra na tela em qual passo o algoritmo se encontra e o que está executando a cada momento.

4. VALIDAÇÃO E RESULTADOS

Esta seção aborda a validação e os resultados tendo em vista o trabalho proposto. Assim, com intuito de comprovar o funcionamento do modelo implementado, foi escolhida a cidade de Mossoró-RN como estudo de caso. Os testes foram realizados com duas instâncias de características diferentes. A primeira é uma instância formada por coordenadas aleatórias de vários pontos da cidade, já a segunda é uma instância constituída de localizações mais específicas, posicionadas no Bairro Belo Horizonte e nos seus arredores.

4.1 Metodologia dos testes

Primeiramente realizou-se um teste unitário, onde cada classe foi testada separadamente, buscando descobrir as falhas e comprovar o funcionamento das entradas. Após testar cada parte da implementação separadamente, o segundo teste teve como objetivo verificar o funcionamento do modelo como um todo, com cada etapa servindo de entrada para a próxima. Neste primeiro teste optou-se por compor instâncias com coordenadas aleatórias e dispersas pois dessa forma observa-se o funcionamento do algoritmo quando os pontos eram separados por distâncias maiores.

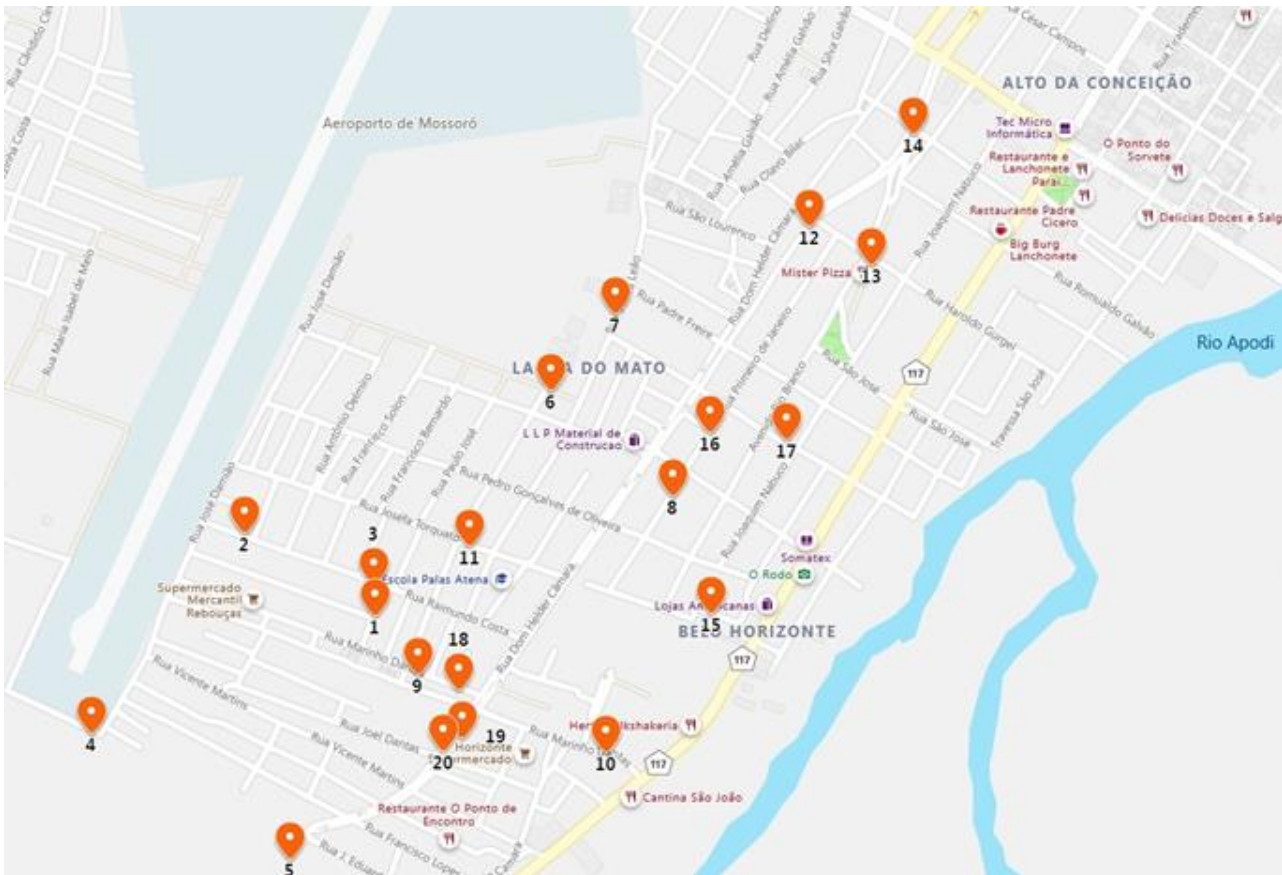
O segundo teste auxiliou na observação dos resultados que o algoritmo consegue gerar com uma instância de pontos mais próximos, onde as distâncias entre os vértices são pequenas. Além disso, essa segunda instância foi desenvolvida para que o resultado dispusesse de rotas prontas para a coleta, já que se refere a pontos reais.

4.2 Resultados

A metodologia de coleta das localizações envolveu o uso do sistema de satélite do Google *Maps*, utilizado para observação das ruas do bairro Belo Horizonte e possíveis pontos de descarte irregular. Considerou-se como um ponto válido as localizações que possuíam descartes de resíduos sólidos em áreas públicas e irregulares. Normalmente os pontos são lugares onde os moradores daquela vizinhança depositam lixo e o serviço de limpeza urbana não coleta na sua passagem.

Devido ao satélite possuir apenas imagens das ruas dos anos de 2011/2012, foi realizada uma visita em alguns locais para que se tivesse a confirmação de que ainda existiam resíduos descartados. Com a coleta das coordenadas realizada, montou-se uma instância com vinte localizações de descarte irregular de lixo em áreas públicas. A Figura 11 ilustra os pontos que compõem essa instância.

Figura 11 – Pontos de descarte irregular no bairro Belo Horizonte.



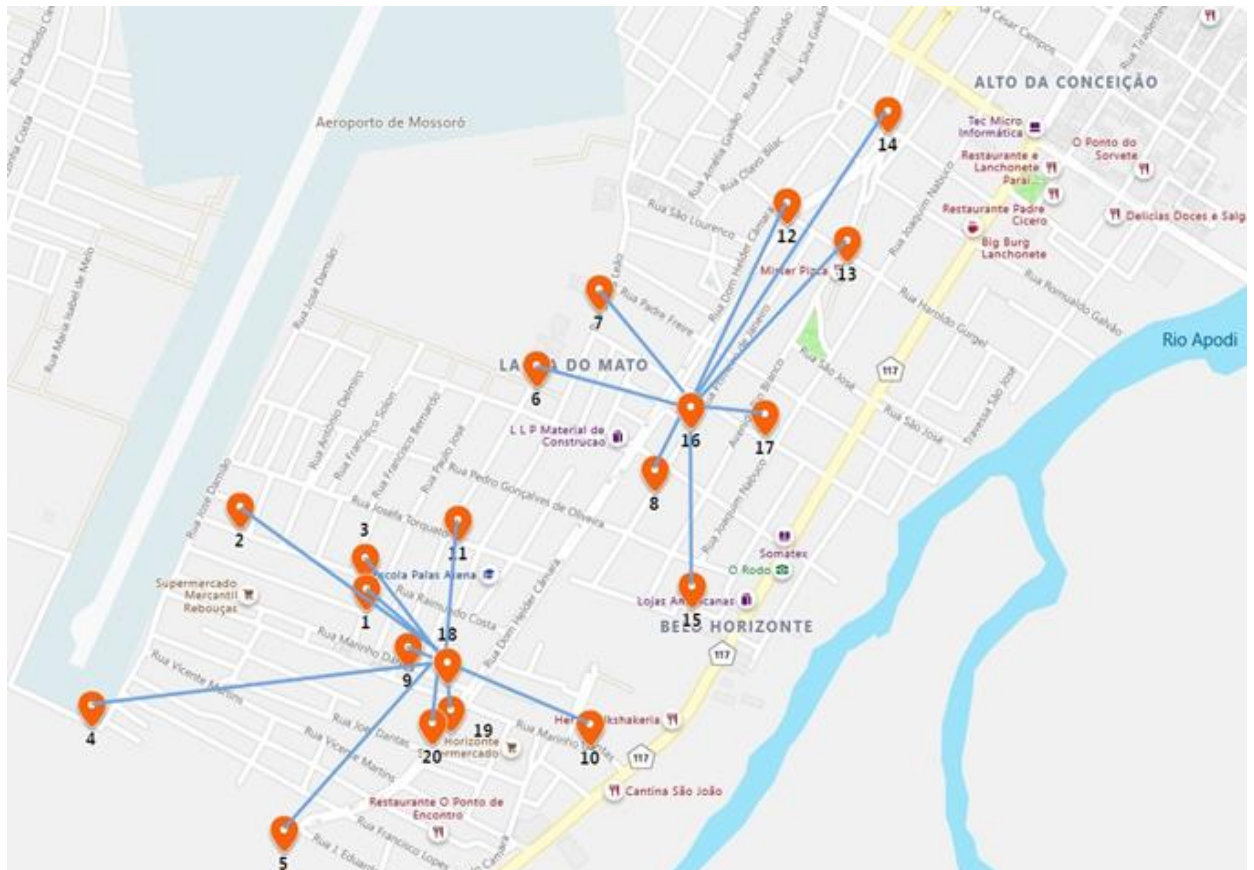
Fonte: Autoria própria.

O primeiro resultado do algoritmo, após a transformação das coordenadas na matriz geral de distâncias, é o cálculo das p-medianas. Por se tratar de uma instância com poucos pontos, se o número de medianas for alto, serão geradas rotas muito pequenas. Diante disto, foram realizadas duas execuções do algoritmo. A primeira para duas medianas e a segunda para três. Os resultados desta etapa são ilustrados na Figura 12 e na Figura 13. O primeiro resultado tem uma soma final de 8,042 quilômetros e o segundo 6,58 quilômetros. O Algoritmo Genético foi executado com os seguintes parâmetros:

- Tamanho da população: 100;
- Quantidade de gerações: 3000;

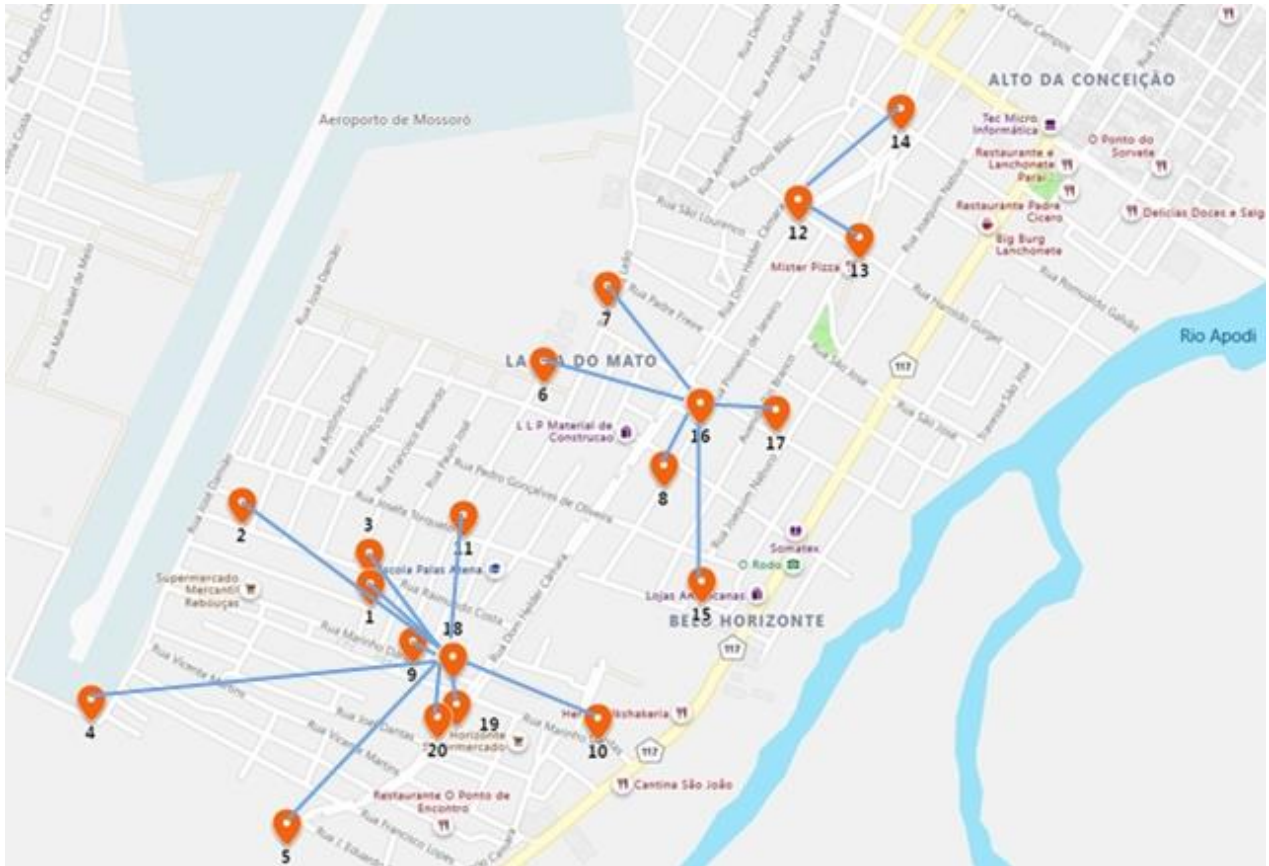
- Taxa de cruzamento: 90%;
- Taxa de mutação: 10%;
- Indivíduos salvos na mutação: 20%;
- Indivíduos salvos na seleção: 50%.

Figura 12 – Instância do bairro Belo Horizonte com duas medianas.



Fonte: Autoria própria.

Figura 13 – Instância do bairro Belo Horizonte com três medianas.



Fonte: Autoria própria.

O segundo resultado (calculado a partir das matrizes geradas pelos AGs) são as rotas calculadas pelo GRASP. Para a primeira execução foram geradas duas rotas com tamanhos de 3,081 quilômetros e 3 quilômetros. As Figuras 14 e 15 foram geradas pelo algoritmo e representam essas rotas.

Já na segunda execução, foram definidas calculadas três rotas: a primeira com tamanho de 467 metros, a segunda com 1,83 quilômetros e a terceira com 3,081 quilômetros. As duas primeiras são apresentadas nas Figuras 16 e 17, que também foram geradas pelo algoritmo e a terceira rota é igual à da Figura 14. O GRASP foi executado com os seguintes parâmetros:

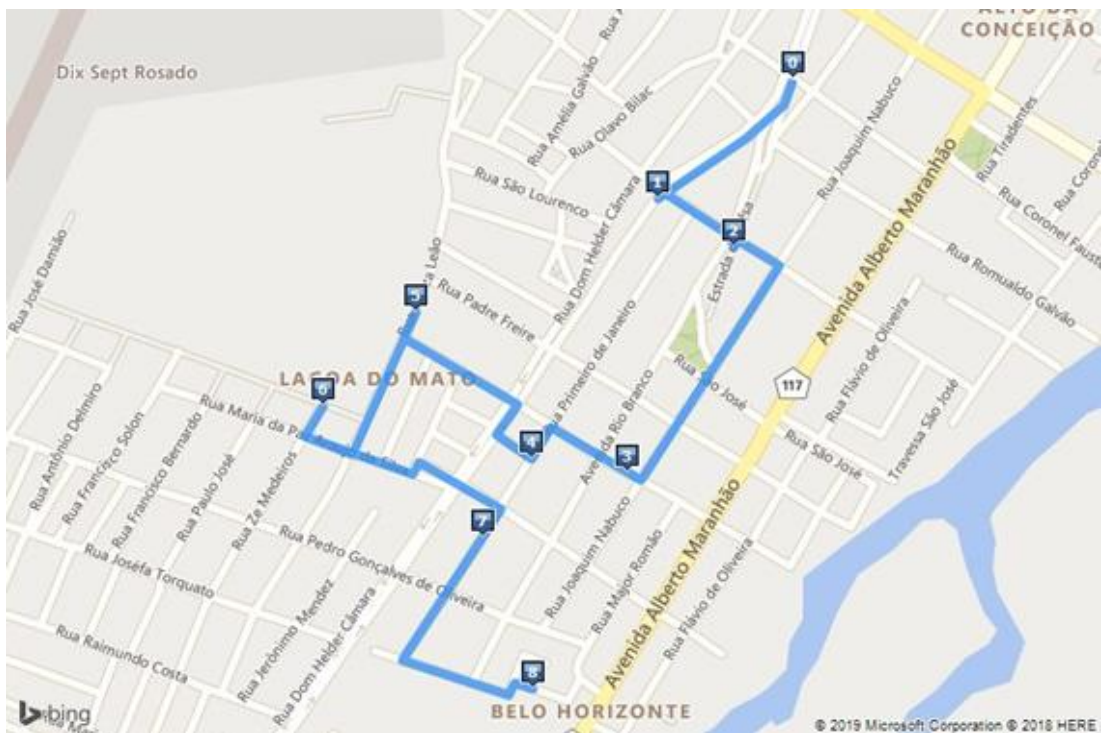
- Alfa: 0.3;
- Quantidade de iterações: 1000.

Figura 14 – Imagem gerada pelo algoritmo com a primeira rota da primeira execução.



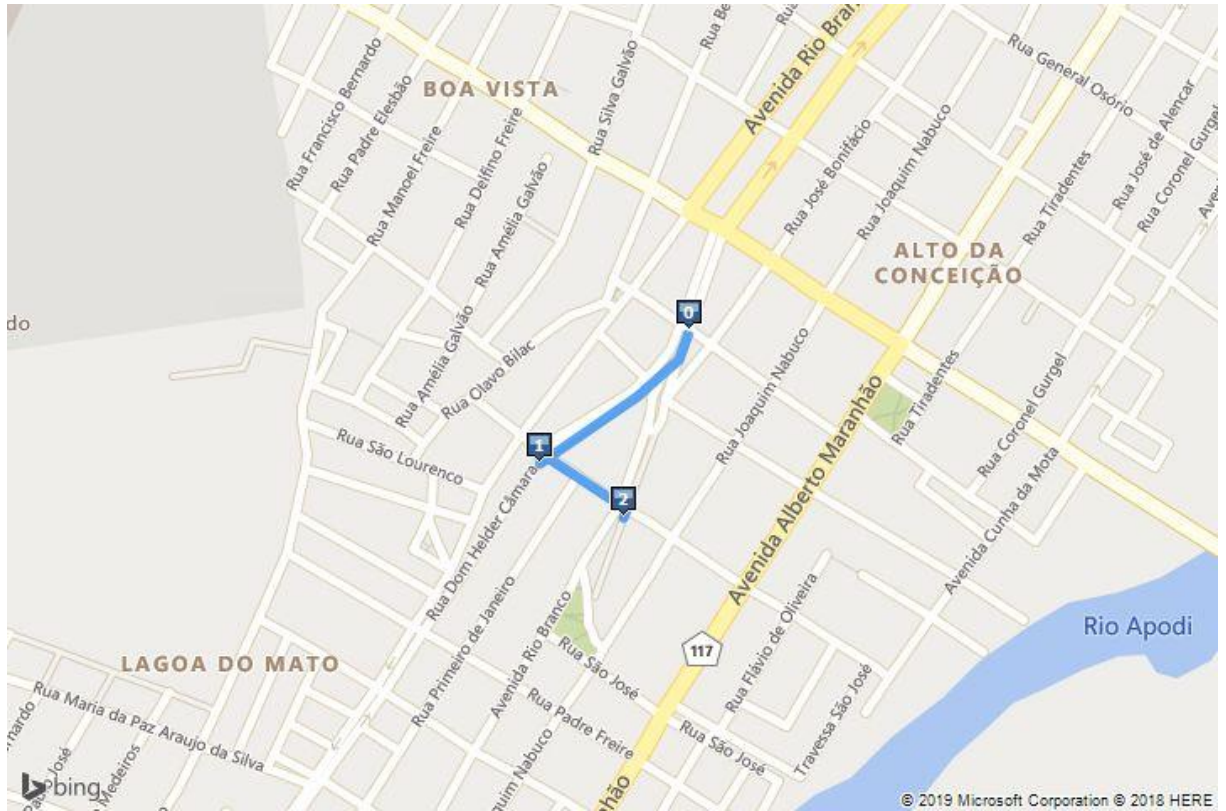
Fonte: Autoria própria

Figura 15 – Imagem gerada pelo algoritmo com a segunda rota da primeira execução.



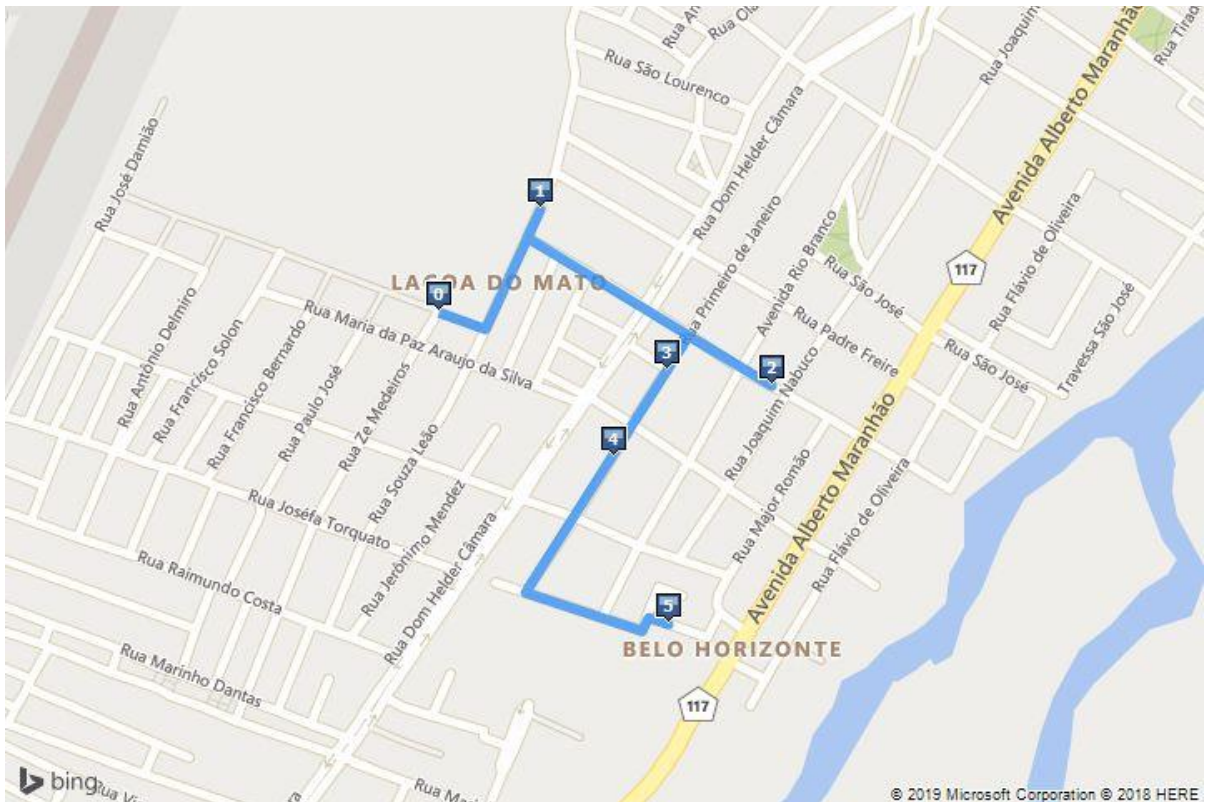
Fonte: Autoria própria

Figura 16 – Imagem gerada pelo algoritmo com a primeira rota da segunda execução.



Fonte: Autoria própria

Figura 17 – Imagem gerada pelo algoritmo com a segunda rota da segunda execução.



Fonte: Autoria própria

Com a implementação do método proposto torna-se possível auxiliar no gerenciamento do roteamento de veículos para coleta de lixo de descarte irregular no Bairro de Belo Horizonte na cidade de Mossoró. Por conseguinte, facilitando que outras áreas desta mesma cidade possam passar pelo processo de otimização das rotas de veículos assim contribuindo com a diminuição dos custos gerados para coleta de lixo.

5. CONSIDERAÇÕES FINAIS

Diante do atual cenário das cidades após um estudo sobre descarte do lixo em lugares irregulares, observou-se a importância do Gerenciamento Integrado de Resíduos Sólidos Urbanos e que são necessárias soluções para obter melhorias na vida da população. Desta forma, constatou-se a utilidade de otimizar o roteamento de veículos para coleta de lixo e o gerenciamento de etapas integradas, assim concebeu-se uma etapa que se baseia na utilização de recursos tecnológicos no processo, tendo em vista auxiliar os gestores nas tomadas de decisão com objetivo principal de diminuir os gastos.

Diante disso, este trabalho tem por objetivo geral o desenvolvimento de um método de otimização de roteamento de veículos para a coleta de lixo de descarte irregular em área pública, com a utilização de algoritmos de otimização. Assim, implementou-se um algoritmo utilizando a linguagem Python, realizando a transformação dos dados geográficos em matrizes de adjacência, para facilitar a representação dos dados em forma de grafos. Além disso, aplicou-se uma metaheurística GRASP e um AG ao problema de Roteamento de Veículos, determinando medianas que serviram para dividir o grafo em grupos e determinando uma rota otimizada para cada veículo.

O modelo foi efetivado através dos testes dos algoritmos para validação utilizando Mossoró-RN como estudo de caso especificamente o Bairro Belo Horizonte e os seus arredores, resultando na produção de imagens com as rotas para uma melhor visualização.

O Algoritmo Genético aplicado ao problema das p-medias, por mais que hajam vários parâmetros e que a representação de um indivíduo não seja tão trivial, cumpriu os objetivos, conseguindo gerar bons pontos de facilidade. Já o algoritmo GRASP aplicado ao cálculo de rotas de menor caminho também se mostrou eficiente.

Dado o exposto nesta pesquisa, como perspectivas futuras deste trabalho, têm-se:

- Conseguir informações sobre os pontos de lixo assim como sobre os veículos de coleta e aplicar os conceitos de CVRP;

- Desenvolver uma plataforma para que os cidadãos possam relatar o descarte irregular de lixo nas suas ruas para geração de novas instâncias;
- Realizar estudo sobre a complexidade dos algoritmos desenvolvidos;
- Desenvolver e aplicar outras metaheurísticas aos problemas;
- Implementar uma interface gráfica para o *software*.

REFERÊNCIAS

ALP, O; ERKUT, E. An eficiente Genetic Algorithm for the p-median problem. *Annals of Operational Research* 122, 2003.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS – ABNT. NBR ISO/IEC 10004/2004: Resíduos Sólidos - Classificação. Rio de Janeiro. 2004.

BENJAMIN, A.M; BEASLEY, J.E. Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. *Computers & Operational Research*. Volume 37, 2010.

BRASIL. Lei nº 12.305, de 2 de agosto de 2010. Institui a Política Nacional de Resíduos Sólidos; altera a Lei no. 9.605, de 12 de fevereiro de 1998; e dá outras providências. Disponível em: <<http://www2.mma.gov.br/port/conama/legiabre.cfm?codlegi=636>>. Acesso em 19/03/2019

BRITO et al. A GRASP-VNS hybrid for the fuzzy vehicle routing problem with time windows. *Lectures Notes in Computer Science*. Springer, 2013.

BUHRKAL, K; LARSEN, A. ROPKE, S. The waste collection vehicle routing problem with time windows in a city logistics context. *The Seventh International Conference on City Logistics*. Kogens Lyngby, 2012.

CAVALCANTE S., FRANCO M. F. A. Profissão perigo: percepção de risco à saúde entre catadores do Lixão do Jangurussu. *Revista Mal-Estar e Subjetividade*, Fortaleza, vol. 7, n. 1, p. 211-231, 2007.

CHAVES, A. A. Uma meta-heurística híbrida com busca por agrupamentos aplicada a problemas de otimização combinatória. Tese de doutorado em Computação Aplicada, Universidade Federal de São Paulo, 2009.

COLIN, E. C. Pesquisa Operacional: 170 aplicações em estratégia, finanças, logística, produção, marketing e vendas. Rio de Janeiro, 2007.

CRAINIC et al. GRASP with path relinking for the two-echelon vehicle routing problem. *Operations Research/Computer Science Interface Series*. Volume 53. Springer. Nova Iorque, 2013.

EXAME. Brasil diz adeus ao maior lixão da América Latina, em Brasília. 2018. Disponível em <<https://exame.abril.com.br/brasil/brasil-diz-adeus-ao-maior-lixao-da-america-latina-em-brasilia/>>. Acesso em: 19/03/2019.

EXAME. Quanto lixo os brasileiros geram por dia em cada estado. 2016. Disponível em < <https://exame.abril.com.br/tecnologia/quanto-lixo-os-brasileiros-geram-por-dia-em-cada-estado/>>. Acesso em 19/03/2019.

FEO, T.A; RESENDE, M.G.C. A probabilistic heuristic for a computationally difficult set covering problem. *Operation Research Letters*. 1989.

FOLHA DE SÃO PAULO. Descarte de lixo ainda é inadequado em mais da metade do país. 2018. Disponível em <<https://www1.folha.uol.com.br/cotidiano/2018/01/1951986-descarte-de-lixo-ainda-e-inadequado-em-mais-da-metade-das-cidades-do-pais.shtml>>. Acesso em: 19/03/2019.

GOLDEN, B.; RAGHAVAN, S.; WASIL, E. *The Vehicle Routing Problem: latest advances and new challenges*. Springer. Nova Iorque, 2008.

GOLDENGORIN, B.; KRUSHINSKY, D.; PARDALOS, P. M. *Cell formation in industrial engineering: theory, algorithms and experiments*. Springer. Nova Iorque, 2013.

HOLLAND, J.H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence (complex adaptive systems)*. The MIT press. Massachusetts, 1975.

ISLER, C. A.; BONASSA, A. C.; CUNHA, C. B. Algoritmo genético para resolução do problema de p-mediana capacitado associado à distribuição de peças automotivas. *Revista Transportes*, 2012.

KRAMER, O. *Genetic algorithms essentials*. Springer. Gewerbestrasse, 2017.

Lixo Municipal: manual de gerenciamento integrado. São Paulo. 4^o ed. 2018

MIYAZAWA, F. K.; SOUZA, C. C. *Introdução a otimização combinatória*. Jornadas de Atualização em Informática – Congresso da Sociedade Brasileira de Computação. 2015

PEREIRA, M. A. Um método Branch-and-Price para problemas de localização de p-mediana. Tese de Doutorado. INPE, São José dos Campos, 2005.

PROGRAMA DE PESQUISA EM SANEAMENTO BÁSICO – PROSAB. Resíduos sólidos urbanos: aterro sustentável para municípios de pequeno porte / coordenador Armando B. de Castilhos Junior. Florianópolis, 2003.

RAMAKRISHNAIAH, C.R; SHIJU, E.J; VATHSALA S. Optimization of Municipal Solid Waste Collection through GIS Technique for Chikmagalur City. *International Journal of Environmental Engineering and Management*. Volume 2. 189-200. (2011).

RESENDE, M.G.C, RIBEIRO, C.C. *Optimization by GRASP: Greedy randomized adaptive search procedures*. Springer. Nova Iorque, 2016.

SOARES, G. F. Algoritmos primais e duais para o problema das p-medianas. Dissertação de mestrado, Pontifícia Universidade Católica, 2009.

TOTH, P.; VIGO, D. The Vehicle Routing Problem. 2002.

YANG et al. Metaheuristic Applications In Structures and Infrastructures. Elsevier. Massachusetts, 2013.

YANG, X. Engineering Optimization: an introduction with metaheuristic applications. Wiley. Nova Jersey, 2010.