



UNIVERSIDADE FEDERAL DO TOCANTINS
CAMPUS DE PALMAS
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

LUIZ FELIPE ROCHA MOREIRA

**APLICAÇÃO DE ALGORITMO GENÉTICO EM UM
PROBLEMA DA ENGENHARIA LOGÍSTICA**

Palmas/TO
2019

LUIZ FELIPE ROCHA MOREIRA

**APLICAÇÃO DE ALGORITMO GENÉTICO EM UM
PROBLEMA DA ENGENHARIA LOGÍSTICA**

Trabalho de Conclusão de Curso de graduação II foi avaliada e apresentada à UFT – Universidade Federal do Tocantins – Campus Universitário de Palmas, Curso de Engenharia Elétrica para obtenção do título de bacharel e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: Dr. Jadiel Caparrós da Silva

Palmas/TO
2019

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da Universidade Federal do Tocantins

- M838a Moreira, Luiz Felipe Rocha.
 APLICAÇÃO DE ALGORITMO GENÉTICO EM UM PROBLEMA DA
 ENGENHARIA LOGÍSTICA. / Luiz Felipe Rocha Moreira. – Palmas, TO, 2019.
 60 f.
- Artigo de Graduação - Universidade Federal do Tocantins – Câmpus
 Universitário de Palmas - Curso de Engenharia Elétrica, 2019.
 Orientador: Jadiel Caparrós da Silva
 Coorientadora : Marilene Andreia Mantovani
1. Inteligência Artificial. 2. Algoritmo Genético. 3. Logística. 4. Problema
 da Mochila. I. Título

CDD 621.3

TODOS OS DIREITOS RESERVADOS – A reprodução total ou parcial, de qualquer forma ou por qualquer meio deste documento é autorizado desde que citada a fonte. A violação dos direitos do autor (Lei nº 9.610/98) é crime estabelecido pelo artigo 184 do Código Penal.

Elaborado pelo sistema de geração automática de ficha catalográfica da UFT com os dados fornecidos pelo(a) autor(a).



ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO – TCC - 2

No *dia 27 do mês de Março de dois mil e dezenove*, o discente Luiz Felipe Rocha Moreira, matriculado sob o nº. 2014110184, tendo como banca examinadora seu orientador, o *Prof. Dr. Jadiel Caparrós da Silva*, e os professores: *Prof^a. Dr^a. Marilene Andreia Mantovani* e o *Prof. Dr. Humberto Xavier de Araujo*, apresentou o Trabalho de Conclusão de Curso intitulado “**APLICAÇÃO DE ALGORITMO GENÉTICO EM UM PROBLEMA DA ENGENHARIA LOGÍSTICA**” obtendo a nota:

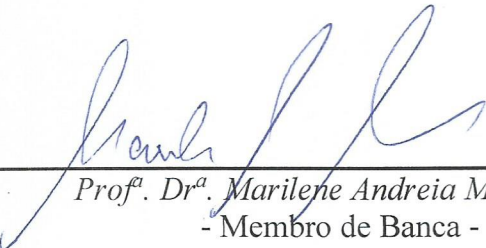
9 (nove) e conceito (Aprovado).



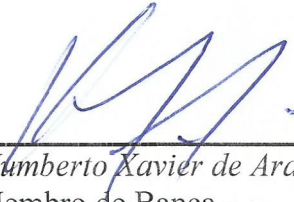
Prof. Dr. Jadiel Caparrós da Silva
- orientador -



Luiz Felipe Rocha Moreira
- discente -



Prof. Dr^a. Marilene Andreia Mantovani
- Membro de Banca -



Prof. Dr. Humberto Xavier de Araujo
- Membro de Banca -

LUIZ FELIPE ROCHA MOREIRA

**APLICAÇÃO DE ALGORITMO GENÉTICO EM UM PROBLEMA
DA ENGENHARIA LOGÍSTICA**

Trabalho de Conclusão de Curso de graduação II foi avaliada e apresentada à UFT – Universidade Federal do Tocantins – Campus Universitário de Palmas, como requisito para aprovação em Projeto de Graduação I do curso de Engenharia Elétrica.

Professor Orientador Dr. Jadiel Caparrós da Silva.

Prof. Dr. Jadiel Caparrós da Silva
Universidade Federal do Tocantins

Prof^ª. Dra. Marilene Andreia Mantovani
Universidade Federal do Tocantins

Prof^ª. Dr. Humberto Xavier de Araujo
Universidade Federal do Tocantins

Palmas/TO
2019

RESUMO

O trabalho de pesquisa descrito nesse documento tem o intuito de desenvolver um Algoritmo Genético com viés de aplicação na resolução de um problema da Engenharia Logística. Tendo esse objetivo claro a composição da pesquisa será em um primeiro momento o esclarecimento dos conceitos técnicos utilizados ao longo do projeto. Após isso é apresentado o problema que será solucionado assim como sua modelagem, para que ao final seja apresentado um algoritmo, focando na construção da lógica do algoritmo ao invés dos conceitos relativos a linguagem de programação escolhida, assim como os dados obtidos que comprovem a eficácia da ferramenta como forma de solução para o problema definido. Os conceitos por trás do algoritmo aqui utilizado, derivam dos estudos mais recentes sobre Inteligência Artificial e tem sua fundamentação nos estudos da biologia da teoria da evolução e genética.

PALAVRAS-CHAVE: Inteligência Artificial, Algoritmo Genético, Crossover, Logística, Problema da Mochila.

ABSTRACT

The research work described in this document intends to develop a Genetic Algorithm with application bias in solving a Logistic Engineering problem. Having this clear objective the composition of the research will be in the first moment the clarification of the technical concepts used throughout the project. After this, the problem will be solved, as well as its modeling, so that in the end an algorithm is presented, focusing on the construction of the logic of the algorithm instead of the concepts related to the chosen programming language, as well as the obtained data that prove the effectiveness of the tool as a solution to the defined problem. The concepts behind the algorithm used here derive from the most recent studies on Artificial Intelligence and have their basis in the studies of the biology of evolutionary theory and genetics.

KEY WORDS: Artificial Intelligence, Genetic Algorithm, Crossover, Logistics, Knapsack Problem.

Índice de Figuras

| | |
|---|----|
| Figura 1 - Diagrama que posiciona os algoritmos evolucionários como técnica de busca. | 20 |
| Figura 2 - Esquema de um algoritmo genético..... | 27 |
| Figura 3 – Algoritmo Genético Abstrato..... | 29 |
| Figura 4 - Definição da classe Produto (Abstração)..... | 30 |
| Figura 5 - Definição da classe Individuo (Abstração)..... | 31 |
| Figura 6 – Função Avaliação (Abstração)..... | 33 |
| Figura 7 - Cruzamento de um ponto..... | 34 |
| Figura 8 – Operador de Mutação..... | 35 |
| Figura 9 – Organograma do AG montado..... | 37 |
| Figura 10 – Ilustração do método da roleta-viciada..... | 38 |
| Figura 11 – Escopo do AG utilizado (Abstração)..... | 40 |
| Figura 12 – Solução para o caso P01 encontrada pelo AG..... | 45 |
| Figura 13 – Gráfico da solução para o caso P01 encontrada pelo AG..... | 45 |
| Figura 14 – Solução para o caso P01 encontrada pelo AG..... | 46 |
| Figura 15 – Gráfico da solução para o caso P01 encontrada pelo AG..... | 47 |
| Figura 16 – Solução para o caso montado encontrada pelo AG Teste 1..... | 49 |
| Figura 17 – Gráfico da solução para o caso montado encontrada pelo AG Teste 1..... | 49 |
| Figura 18 – Solução para o caso montado encontrada pelo AG Teste 2..... | 50 |
| Figura 19 – Gráfico da solução para o caso montado encontrada pelo AG Teste 2..... | 51 |

Índice de Tabela

| | |
|--|----|
| Tabela 1 - Cronograma de Execução das Atividades | 22 |
| Tabela 1 - Definição do Individuo..... | 30 |
| Tabela 2 – Descrição do Problema P01..... | 43 |
| Tabela 3 – Parâmetros utilizados para P01..... | 44 |
| Tabela 4 – Problema da locação de cargas no caminhão..... | 48 |

SUMÁRIO

| | |
|--|-----------|
| 1. INTRODUCAO | 8 |
| 1.1. CONSIDERAÇÕES INICIAIS | 8 |
| 1.2. JUSTIFICATIVA..... | 9 |
| 1.3. OBJETIVOS | 10 |
| 1.3.1. OBJETIVO GERAL..... | 10 |
| 1.3.2. OBJETIVOS ESPECÍFICOS | 10 |
| 1.4. ESTRUTURA DO TRABALHO..... | 10 |
| 1.5. COMENTÁRIOS..... | 11 |
| 2. O PROBLEMA DA MOCHILA | 12 |
| 2.1. CONSIDERAÇÕES INICIAIS..... | 12 |
| 2.2. DEFINIÇÃO DO PROBLEMA | 14 |
| 2.2.1 Definição teórica..... | 15 |
| 2.3. COMENTÁRIOS..... | 18 |
| 3. ALGORITMOS GENÉTICOS | 19 |
| 3.1. CONSIDERAÇÕES INICIAIS | 19 |
| 3.2. ALGORITMOS EVOLUCIONÁRIOS | 19 |
| 3.3. ALGORITMOS GENÉTICOS | 20 |
| 3.3.1. Definição computacional | 21 |
| 3.3.2. Características de um AG | 22 |
| 3.3.3 Elementos Básicos de um AG..... | 22 |
| 3.3.4. Processamento de um AG | 23 |
| 3.3.5. Convergência Genética e Operador de Mutação | 24 |
| 3.4. COMENTÁRIOS..... | 25 |
| 4. METODOLOGIA..... | 26 |
| 4.1. INTRODUÇÃO AO PROBLEMA E A FERRAMENTA..... | 26 |

| | |
|---|-----------|
| 4.2. O PROBLEMA A SER TRATADO | 27 |
| 4.3 MODELAGEM | 28 |
| 4.3.1 Modelagem do Indivíduo | 29 |
| 4.3.2 Modelagem da Avaliação (<i>Fitness</i>) | 33 |
| 4.3.3 Modelagem do Operador de Cruzamento | 34 |
| 4.4.4 Modelagem do Operador de Mutação..... | 35 |
| 4.4 CONSTRUÇÃO DO ALGORITMO..... | 36 |
| 4.4.1 Método “Seleciona_pais” | 37 |
| 4.4.2 Abstração do AG..... | 39 |
| 4.3. COMENTÁRIOS..... | 41 |
| 5. RESULTADOS E DISCUSSÕES..... | 42 |
| 5.1. INTRODUÇÃO | 42 |
| 5.2. ELABORAÇÃO DOS TESTES | 42 |
| 5.3. TESTE 1 – Problema P01..... | 43 |
| 5.4. TESTE 2 – Carregamento de Caminhão | 48 |
| 5.5. COMENTÁRIOS..... | 53 |
| 6. CONCLUSÃO..... | 54 |
| 6.1. TRABALHOS FUTUROS | 55 |
| Referência Bibliográfica..... | 56 |
| APÊNDICE A – PUBLICAÇÕES | 58 |

1. INTRODUCAO

1.1. CONSIDERAÇÕES INICIAIS

O desenvolvimento da ciência tem acompanhado a humanidade desde seus primeiros registros rupestres e o advento do fogo e, desde então, jamais cessou motivado pelo interesse em novas descobertas e o desejo de poder controlar o mundo a sua volta. Tantas eras se passaram e o pensamento científico do homem só ganhou mais e mais espaço e reconhecimento da sociedade, e sua vontade por modificar e entender o ambiente em que vive o fez alcançar avanços muito importantes em diversas áreas como a da saúde, do transporte, da comunicação, entre outras.

No texto de Ribeiro (2000) a história da humanidade pode ser compreendida ao se observar o histórico de evoluções tecnológicas que, dentro de um ciclo, são responsáveis por fazer com que o homem saia de uma condição habitual para uma nova forma de interagir com ele mesmo e/ou o meio a sua volta. O entendimento da evolução da tecnologia humana está, portanto, intimamente ligado a criação de algo que seja objeto de mudança de um hábito que passa a ser realizado de outra maneira ou visto de outra forma através de um novo conhecimento obtido.

Após muitos anos de estudo, a ciência possibilitou a criação e utilização de ferramentas que auxiliassem a humanidade a solucionar certos problemas específicos como: realizar cálculos matemáticos, observar os astros no céu e perceber organismos não visíveis a olho nu. Estes foram trabalhos direcionados a calculadoras, telescópios e microscópios, todos equipamentos criados pelo homem que cumprissem, de algum modo, uma tarefa à qual o próprio corpo e intelecto humano não conseguiriam, ao menos não em um curto espaço de tempo.

Com o passar das eras, a relação de instrumento entre homem e máquina ganhou novos conceitos, o objetivo de tais ferramentas passou a ser mais complexo tornando-se objeto de estudo para que se desenvolvesse algo que, agora, não operasse mais simplesmente desenvolvendo alguma habilidade humana, mas, agindo como um ser humano na resolução de um problema.

Nesse âmbito pode-se conceituar todo e qualquer algoritmo criado até hoje, contudo, os que mais se aproximam dessa objetificação dos estudiosos são os algoritmos evolucionários, mais especificamente os genéticos. Segundo Linden (2012, p. 46) “Algoritmos genéticos (AGs)

são um ramo dos algoritmos evolucionários e como tal podem ser definidos como uma técnica de busca baseada numa metáfora do processo biológico de evolução natural.”

Através dessa ferramenta a solução de problemas conhecidos como problemas intratáveis ganharam uma forma de resolução. Os problemas intratáveis são assim nomeados devido ao tempo necessário para sua solução serem de escala inaceitável para um ser humano resolve-lo por meios comuns. Esses problemas estão impermeados em processos básicos do cotidiano, em sua maioria imperceptíveis para um observador leigo no assunto.

Contudo, tais problemas compõem questões fundamentais de algumas tecnologias utilizadas atualmente como sites de busca, gerenciadores de sistemas logísticos, softwares de geolocalização, entre outros. Tais ferramentas se tornaram peças importantes na execução e funcionamento, de várias áreas da sociedade, como os estudantes que utilizam o AG dos sites de busca ou as empresas de distribuição de alimentos que utilizaram algum AG para otimizar sua rede logística.

O intuito desse trabalho é calcado nessa necessidade de otimização dos processos relacionados a logística e suas áreas de abrangência, utilizando para isso técnicas da heurística que possuem um potencial de otimização global, com o potencial de encontrar uma solução considerada ótima para o problema.

1.2. JUSTIFICATIVA

A logística é atualmente parte integrante fundamental da máquina que move a sociedade, as rotinas e os processos que os cercam. De maneira não perceptível, a logística desenvolve um importante papel na sociedade proporcionando maior facilidade do acesso aos produtos do cotidiano dos consumidores.

Nesse aspecto logístico, o simples processo de transporte de cargas pode ser minuciosamente analisado para que uma solução de otimização possa ser projetada e implementada no intuito de gerar uma economia de tempo e recursos gastos. Dentro dessa problemática, é notável o esforço da comunidade científica em prol do desenvolvimento de novas ferramentas que possibilitem a otimização dos processos logísticos através de técnicas de Inteligência Artificial.

Dentre este conceito, este trabalho propõe a utilização dos AGs como ferramenta de solução dos problemas enfrentados pelo setor logístico que aparentemente são simples. No entanto, é necessário um grande esforço para determinar, por exemplo, a melhor localização de um centro de distribuição (CDD) que possui um determinado número de localidades a atender,

possuindo um determinado número e configuração específicas de rotas para essa tarefa. A percepção da complexidade que o problema toma ao se trabalhar com um grande número de localidades, se for realizada a tentativa de utilização de uma solução exata, faz com que o uso do AG seja tomado como a melhor solução do problema, trazendo uma resposta em tempo aceitável, eficiente e otimizada ao problema.

1.3. OBJETIVOS

1.3.1. OBJETIVO GERAL

Utilizar uma ferramenta computacional, em uma linguagem de programação de nome Python, utilizando AG aplicados na solução do problema da mochila, com as devidas adaptações de modo a se tornar semelhante a um problema envolvendo o carregamento de um caminhão. A aplicação tem por intuito otimizar os processos de logística de forma a gerar economia de tempo e recursos financeiros de uma situação montada que simula um caso real.

1.3.2. OBJETIVOS ESPECÍFICOS

Esse trabalho tem como objetivo específico:

- Demonstração do funcionamento de um AG básico, além da sua aplicação ao problema NP-Hard do problema da mochila onde serão usados conhecimentos mais refinados.
- Apresentar a composição da ferramenta em uma linguagem de programação atual.
- Gerar uma solução que venha a ser útil quando aplicado em situações reais, e
- Contribuir para o desenvolvimento da otimização de processos das áreas da engenharia.

1.4. ESTRUTURA DO TRABALHO

A estrutura textual do trabalho é dividida em 6 capítulos.

- ✓ Capítulo 1: Apresentar o contexto que justifica o trabalho realizado, apresentando teoricamente o problema e a ferramenta utilizada como solução.
- ✓ Capítulo 2: Apresentar o problema do NP-Hard do problema da mochila que é análogo ao problema que será modelado.

- ✓ Capítulo 3: Elucidar todos os conceitos a respeito dos Algoritmos Genéticos e suas técnicas que serão utilizadas.
- ✓ Capítulo 4: Discorrer sobre a ordem dos processos que levarão a confecção do algoritmo, desde a modelagem do problema até a apresentação dos resultados.
- ✓ Capítulo 5: Apresentar os resultados obtidos através das experimentações comentando alguns elementos presentes nas soluções geradas pelo algoritmo.
- ✓ Capítulo 6: Discorrer sobre os resultados encontrados, apresentar a conclusão obtida quanto a eficiência do método usado como ferramenta para solucionar o problema proposto e citar possíveis trabalhos futuros.

1.5. COMENTÁRIOS

Neste primeiro capítulo, buscou-se apresentar os detalhes e a importância do setor de logística para a sociedade bem como os problemas enfrentados pelo mesmo, seguido pelos objetivos deste trabalho, sua proposta e a organização do trabalho.

2. O PROBLEMA DA MOCHILA

2.1. CONSIDERAÇÕES INICIAIS

Problemas intratáveis ou demasiado complexos devidos a sua enorme quantidade de dados são comuns na natureza e nas áreas da sociedade. Problemas desse caráter podem ser classificados em duas tipologias principais: fáceis e difíceis, sendo definidos dentro do trabalho de Almeida (2010) como:

- Problemas “fáceis”: quando a solução do problema pode ser encontrada fazendo uso de algoritmos polinomiais.
- Problemas “difíceis”: são problemas onde suas únicas formas de solução conhecidas para resolvê-los são algoritmos exponenciais.

Essa classe de denominação indica a complexidade do tempo necessário para que seja encontrada a solução para o problema, utilizando como método alguma técnica de “força bruta”, que seria uma metodologia que busca em todo o espaço de soluções a melhor solução. Pode ser definida uma função que determina a complexidade para as duas tipologias de problemas, Problemas “fáceis” (PF) e Problemas “difíceis” (PD), sendo respectivamente:

Função de Complexibilidade PF = $S(p(n))$, onde $p(n)$ é um polinômio

Para esta função, é notável como a complexibilidade irá variar em grandeza em relação aos PFs. Para todo problema dessa natureza o crescimento ou decrescimento da complexidade será polinomial, como por exemplo: pesquisa binária ($S(\log(n))$), pesquisa sequencial ($S(n)$), ordenação por inserção ($S(n^2)$), e multiplicação de matrizes ($S(n^3)$).

Função de Complexibilidade PD = $S(m^n)$, onde $n > 1$

Já para a função de complexibilidade de PDs o grau de complexibilidade do problema irá variar de maneira exponencial e cumulativa. Como exemplo mais comum de problema dessa natureza pode ser citado o do Caixeiro Viajante (PCV) que tem função de complexibilidade igual a ($S(n!)$). O crescimento fatorial da complexidade dessa função demonstra que o

problema é inviável de ser solucionado por métodos comuns, mesmo para uma quantidade de dados moderada.

Dentro dessas classificações, feitas no intuito de segregar alguns problemas de natureza estatística em diferentes grupos, existem as classes de problemas P, NP e NP-Completo. Os problemas classificados como P são aqueles cuja solução pode ser encontrada em tempo polinomial, são problemas considerados tratáveis pois sua solução pode ser encontrada em tempo viável. Os problemas NP vêm do inglês *Non-deterministic Problems*, e essa denominação classifica os problemas em que a determinação da solução mais adequada ao problema é inviável, sendo, portanto, apenas verificáveis em tempo polinomial. Essa classificação enquadra os problemas tidos como NP em um grupo onde apenas é possível verificar se uma dada entrada corresponde a uma resposta viável ao problema, em tempo considerado possível.

Os problemas classificados como NP-Completo possuem uma característica que fazem com que a seguinte condição seja obedecida: se uma das instâncias do problema puder ser resolvida, então todo o problema pode ser solucionado em tempo polinomial, diferente do que acontece com o problema NP. Essas características diferem os problemas estatísticos em relação aos métodos utilizados em sua solução, e também conforme o comportamento desses problemas frente a diferentes análises.

Para melhor exemplificar os problemas NP pode ser utilizado o Problema do Caixeiro Viajante (PCV). O PCV é um problema muito comum na área de estudos da análise combinatória e que desperta grande interesse da comunidade científica devido a suas particularidades. O PCV gera esse interesse pois é de fácil descrição e representação, possui uma enorme gama de aplicação e associação à problemas atuais, contudo, é de difícil resolução por métodos comuns. Segundo Luciana Saete Buriol (2000):

O PCV pertence à classe de problemas NP-difíceis, ou seja, o tempo gasto para resolve-lo pode ser exponencial em relação ao tamanho da instância. Devido a isso, a resolução do problema utilizando métodos heurísticos ganha maior importância, principalmente quando aplicado a instancias grandes do problema.

(BURIOL, 2000, p. 15).

A descrição informal do PCV é a seguinte: um caixeiro viajante deseja passar por um determinado número de cidades, espaçadas por distancias distintas entre si e interligadas por diversas rotas, ele deseja sair de uma cidade X e percorrer todas as cidades uma única vez

retornando até a cidade X, esse deslocamento, porém deve ser realizado pelo menor caminho possível.

De modo mais formal pode-se fazer a representação do problema com um grafo completo $G = (P, D)$, onde P é um conjunto composto pelos n vértices formados pelos pontos que representam a localização de cada cidade, e D é o conjunto de linhas que interligam cada par de cidade i e j , sendo esses pertencentes ao conjunto P. A cada linha está associado um valor v_{ij} que representa o valor da distância decorrente do percurso indo da cidade i até a cidade j .

O problema consiste, em uma analogia simplificada, a uma otimização do percurso executado por um viajante fictício que deseja sair de um determinado ponto com suas mercadorias, afim de passar por todo um conjunto de pontos disponíveis, percorrendo assim uma determinada rota, retornando para o ponto inicial ao fim do trajeto.

Como descrito no trabalho de Buriol (2000) o problema cresce em complexidade de forma exponencial a medida em que os dados que o definem são incrementados. De modo direto, a dificuldade do problema aumenta conforme cresce o número de cidades (vértices do grafo) pelas quais o caixeiro necessita percorrer para realizar seu percurso. Para aplicação em um caso real a grandeza do problema ganha dimensões que, caso for aplicado um método determinístico, sua solução se torna impraticável em um tempo considerado viável.

O caráter do problema trata de justificar a aplicação de um método heurístico como forma de solução, uma vez que, quando aplicadas dimensões que tentem se aproximar de uma situação real, ou seja, quando são fornecidos muitos pontos como possibilidades para que o viajante percorra, métodos comuns se tornam uma opção inviável de solução devido ao tempo gasto por essas formas de resolução.

Todas essas informações referentes ao caráter dos problemas não determinísticos são importantes para a compreensão do problema aqui utilizado como caso de estudo, assim como sua modelagem. O conceito de problemas em que encontrar a melhor solução para o mesmo é impensável para métodos comuns é a motivação da utilização das técnicas que serão abordadas mais à frente.

2.2. DEFINIÇÃO DO PROBLEMA

Neste capítulo serão apresentados os conceitos que definem o problema que será utilizado ao longo do trabalho como base para a modelagem do algoritmo e seus elementos.

Dentro desses conceitos estão a sua definição teórica e matemática que traduzem o caráter do problema para bases que possam ser modeladas e aplicadas a casos reais.

2.2.1 Definição teórica

O Problema da mochila (PM) pode ser definido em teoria da seguinte maneira:

[...] Suponha que um alpinista deva carregar sua mochila de itens, dentre vários disponíveis, considerando a máxima capacidade suportável, que denominamos de capacidade da mochila. A cada item é atribuído um valor de utilidade e o alpinista deve selecioná-los buscando maximizar o valor de utilidade total.

MARQUES (2004, p. 22)

Modelando matematicamente o problema citado podem ser considerados os seguintes dados:

n = número de itens disponíveis;

v_i = valor de utilidade do item, onde $i = 1, 2, \dots, n$;

p_i = peso do item, onde $i = 1, 2, \dots, n$;

L = capacidade da mochila.

Partindo dessa definição surgiram diversas outras denominações e variações para esse mesmo problema, como o Problema da mochila 0-1 e o Problema da mochila inteira, que são os casos que serão utilizados como base na modelagem do algoritmo. Apesar de serem variações do mesmo problema, tanto o Problema da mochila 0-1 quanto o problema da mochila inteira seguem as mesmas premissas da definição básica: existe uma determinada quantidade de itens e estes devem ser alocados em uma mochila de capacidade finita, sobre cada item é atribuído um valor de importância e o intuito do tratamento do problema é maximizar o valor de importância comportado pela mochila.

Seguindo a ordem de citação, o Problema da mochila 0-1 é o caso comum do PM, com a restrição de que é possível selecionar apenas um item de cada um dos disponíveis, mesmo que estes tenham quantidades disponíveis variadas. Segundo Marques (2004) este é um dos problemas de otimização discreta mais estudado, devido a alguns fatores chave como:

1) Pode ser visualizado como o problema de otimização linear inteira mais simples;

2) Aparece frequentemente em outros problemas mais complexos como um subproblema a ser resolvido;

3) Consegue representar um grande leque de situações reais;

4) Qualquer problema de otimização linear inteira pode ser transformado em um problema da mochila 0-1.

5) Aplicação em problemas de otimização combinatória, com viés de aplicação nas áreas da Engenharia Elétrica como nos Sistemas Elétricos de Potência (SEP)

Matematicamente esse problema é composto de uma variável de decisão, definida como x_i , onde caso o produto for selecionado para a mochila a variável receberá valor 1, em caso contrário, do produto não ser selecionado para a mochila, a variável receberá valor 0. Resultando na seguinte notação:

$$x_i = \begin{cases} 1, & \text{se o item } i \text{ for selecionado;} \\ 0, & \text{para o caso contrário.} \end{cases} \quad , \text{ onde } i = 1, 2, \dots, n$$

Para que, através da seleção de cada um dos itens o alpinista possa maximizar o valor atribuído a carga da mochila ele deve seguir o critério de máxima capacidade da mochila, que resulta nas seguintes expressões matemáticas:

$$\text{Função a ser maximizada: } \theta = \sum_{i=1}^n v_i \cdot x_i \quad (1)$$

$$\text{Sob as condições de: } \sum_{i=1}^n l_i \cdot x_i \leq L \quad (2)$$

$$\text{Onde } x_i = 0 \text{ ou } 1 \text{ e } i = 1, 2, \dots, n \quad (3)$$

A modelagem matemática do problema o transforma em um somatório do produto dos valores de utilidade do item pela variável de escolha, que retorna o valor de utilidade total atribuído a mochila ao final da decisão de quais itens levar. Esse somatório, porém, deve obedecer às condições de não ultrapassar a capacidade máxima da mochila, representada pela equação (2) assim como seguir as condições de limitação em que o elemento x_i poderá assumir apenas valor 0 ou 1, e que a variável i vai de 1 até o número de itens disponíveis.

A segunda variação do problema, o Problema da mochila inteira ou apenas Problema da mochila, se assemelha um pouco mais com o exemplo que será utilizado como problema mais à frente no trabalho. A sua definição é a que melhor representa o problema de modo geral, pois,

não há limitações no número de itens que é selecionado. Essa característica faz com que seja permitido selecionar mais do que um mesmo elemento que esteja disponível, o que aproxima o problema de um caso real de locação de uma determinada carga em um caminhão de espaço disponível L por exemplo.

Definindo matematicamente o problema, existirá como no caso anterior, uma variável posta como a variável de decisão:

$$x_i = \text{quantidade de itens do espaço amostral } i \text{ selecionados} \quad (4)$$

, onde $i = 1, 2, \dots, n$

Utilizando a expressão da variável em (4) pode ser modelada a função que define o funcionamento do problema e da busca pela sua solução, considerando as equações abaixo:

$$\text{Função a ser maximizada: } \theta = \sum_{i=1}^n v_i \cdot x_i \quad (5)$$

$$\text{Sob as condições de: } \sum_{i=1}^n l_i \cdot x_i \leq L \quad (6)$$

$$\text{Onde } x_i \geq 0 \text{ e inteiro, } i = 1, 2, \dots, n \quad (7)$$

Nas equações 5 a 7 se encontra a expressão completa para o problema da mochila inteira. O somatório em 5 representa o valor de utilidade de toda a mochila ao final da escolha de itens, seguindo as condições impostas em 6 e 7 que, diferentemente do caso anterior, permitem agora a locação de mais do que um único item da mesma natureza na mochila, sendo sua quantidade agora representada por x_i que pode assumir qualquer valor de número inteiro positivo.

Utilizando os conceitos de NP definidos anteriormente o problema da mochila se encaixa como um problema NP-Completo, tendo seu nível de complexidade definido pela seguinte função de complexibilidade:

$$\text{Função de Complexibilidade } PM = S(p^n), \quad \text{onde } n > 1$$

O problema, portanto, não pode ser solucionado utilizando um algoritmo polinomial, sendo necessária a utilização de outra metodologia não determinística como por exemplo os

métodos heurísticos dos algoritmos evolucionários, que serão a ferramenta escolhida como solução do problema aqui moldado.

Utilizando a analogia do PM que foi descrita acima, será montado um problema fictício onde uma determinada carga, caracterizada por um valor de utilidade e um peso que possuem individualmente, deverá ser alocada em um caminhão de volume disponível L da maneira mais otimizada possível. A situação criada tem como intuito simular um carregamento corriqueiro de uma transportadora, que terá seu processo de locação das cargas no caminhão otimizado pela ferramenta aqui proposta.

2.3. COMENTÁRIOS

Neste capítulo foi apresentado os conceitos iniciais do Problema da Mochila bem como do Problema do Caixeiro Viajante e a definição teórica do Problema que será utilizado neste trabalho. Estes conceitos foram apresentados por causa da importância e necessidade de se compreender os modelos e algoritmos que serão trabalhados nos próximos capítulo.

3. ALGORITMOS GENÉTICOS

3.1. CONSIDERAÇÕES INICIAIS

A reprodução de conceitos encontrados na natureza, através de estudos científicos para a criação de tecnologias que auxiliem na solução de problemas, é uma prática antiga e comum na sociedade. Ao analisar os fenômenos relacionados a genética, através de um olhar matemático e lógico, é possível absorver dos processos de cruzamento e mutação genética uma ferramenta de varredura poderosa para a resolução de problemas, que possuem uma fundamentação na análise combinatória.

Segundo Linden (2012) os AGs são um ramo dos algoritmos evolucionários e como tal podem ser definidos como uma ferramenta de busca que se utiliza de uma metáfora do processo biológico da evolução natural. Essa utilização gera, portanto, ferramentas que podem ser aplicadas nos mais diversos problemas estocásticos.

3.2. ALGORITMOS EVOLUCIONÁRIOS

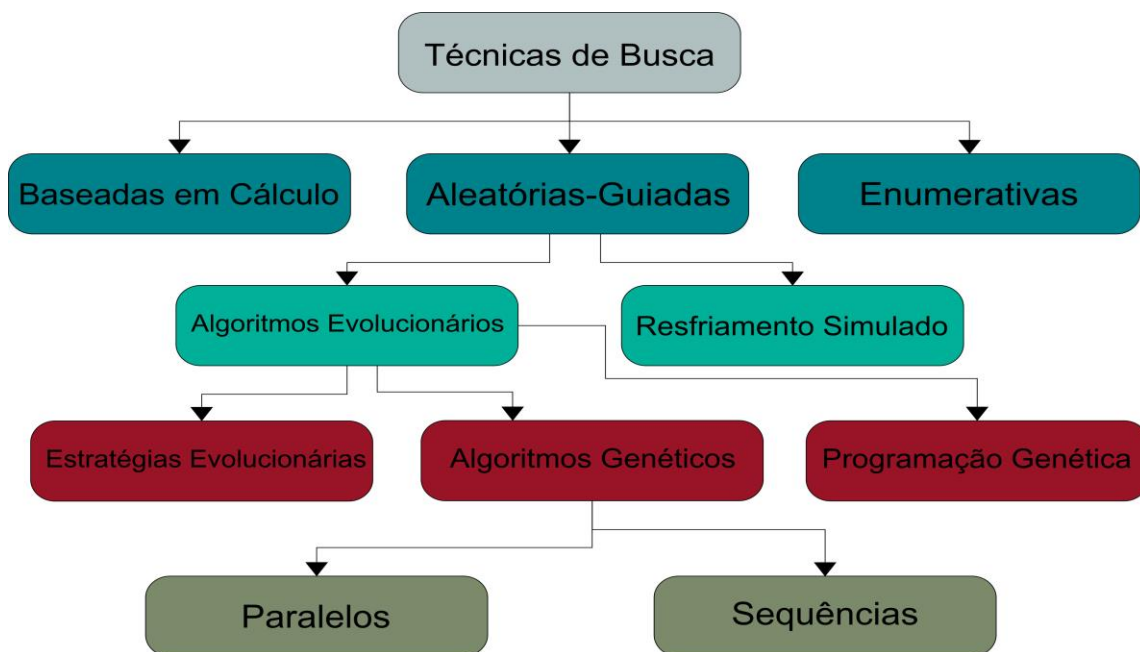
Para elucidar de forma clara as bases que compõem os AGs e seus fundamentos a comparação entre a lógica utilizada e a biologia será constante ao longo do trabalho. A base da técnica utilizada nesse trabalho de conclusão de curso vem dos Algoritmos Evolucionários que são, segundo Linden (2012), modelos computacionais dos processos naturais de evolução como uma ferramenta para resolver problemas.

A analogia ao processo de evolução, advindo da teoria da evolução, a nível de genética está muito presente na produção da lógica dos algoritmos aqui descritos, uma vez que o mesmo atua sobre elementos chamados de indivíduos ou cromossomos, que representam os elementos do problema a ser tratado. Esses por sua vez, estão contidos dentro do espaço amostral do problema, aqui denominados de população. A essas estruturas descritas são aplicados os operadores genéticos, como a recombinação e a mutação que serão os responsáveis por fazer as modificações necessárias sobre os indivíduos que estão distribuídos no espaço amostral criando novos indivíduos. Sobre cada indivíduo é aplicada uma avaliação que irá quantificar a qualidade do indivíduo como solução para o problema proposto. O processo então se repete com a atuação dos operadores genéticos até que a solução encontrada seja satisfatória ou o tempo programado termine.

Os operadores genéticos são a modelagem computacional de fenômenos encontrados na natureza relacionados a genética como a mutação genética, a reprodução sexuada e assexuada, dentre outras modificações genéticas conhecidas pelo universo da biologia.

O sistema descrito é, em sua forma mais primitiva, uma técnica de busca, como representado no organograma da Figura 1, que realiza uma varredura na população ou amostra atual no intuito de encontrar os indivíduos com as melhores características, para então combiná-los através dos operadores genéticos de forma a produzir soluções ainda melhores para o problema, até que as condições de termino sejam correspondidas.

Figura 1- Diagrama que posiciona os algoritmos evolucionários como técnica de busca.



Fonte: Adaptado de LINDEN (2012).

Por se tratar de um sistema que funciona em caráter de loop, é dada a cada nova repetição do algoritmo a denominação de geração. A associação novamente é análoga ao seu equivalente na vida real, uma vez que cabe ao sistema a analogia de dois elementos “Pais” que irão se combinar e gerar elementos “Filhos” formando assim uma nova geração, assim como os elementos “Filhos” irão se combinar com outros elementos dando continuidade ao processo.

3.3. ALGORITMOS GENÉTICOS

Dentro dos conceitos de algoritmo evolucionário outro ramo nasceu, inspirado pelas teorias Neo-Darwinianas, que é a combinação de três fenômenos naturais: evolução das

espécies, seleção natural e herança genética. Dentro dessa teoria, o conceito de evolução está inerente a forma como a vida continua, de modo que sofre a ação de alguns processos no momento da combinação de dois indivíduos de uma população, que provocam variações, sendo eles: reprodução, mutação, competição e seleção (Fogel, 1995). Com base nessas fundamentações é que iniciaram formalmente os estudos a respeito da modelagem computacional desses fenômenos na década de 70.

3.3.1. Definição computacional

A definição absolvida através desses estudos para os AGs os classifica como um método de busca baseado nos fundamentos citados de evolução biológica natural das espécies. Os AGs são classificados também como métodos heurísticos, uma vez que não necessariamente encontram a solução ótima para o problema, e na realidade, quando o fazem, nem sempre conseguiram repetir o feito.

Eles podem ser denominados também como métodos de otimização global, porém possuem grandes divergências desses métodos, tais como o gradiente, uma vez que não estão presos, como o método do gradiente, a buscar, através de uma derivada, o máximo de uma determinada função, o que acaba acarretando a estes métodos não heurísticos uma alienação a resultados encontrados em uma região que pode, ou não, conter a solução mais próxima do ideal para o problema. Ao contrário dessas técnicas os AGs direcionam a busca das melhores soluções do problema dentro de um espaço, utilizando informações históricas para definir novos pontos de busca no intuito de localizar uma região de dados onde são esperados bons resultados (CARVALHO, 2003 apud LINDEN, 2012).

Como método de busca, que tem como alvo a melhor solução de um problema em específico, eles terão como base um ciclo computacional lógico. Esse ciclo será por padrão composto por uma população (universo amostral) sobre a qual atuam operadores genéticos como: mutação e crossover, realizando combinações entre esses indivíduos sobre os quais será imposta uma avaliação baseado no que seria a melhor solução para o problema alvo. Tem-se a repetição dessas etapas até o momento em que se obter uma solução avaliada como boa, podendo ou não ser a melhor solução para o caso.

Os AGs demonstram através desse ciclo de etapas a representação modulada do processo de evolução natural. Uma população inicial qualquer sobre a atuação de processos que combinam suas informações características e geram novos indivíduos que, dentro do espaço amostral e sobre a ação de um outro processo de avaliação, ao final a tendência seja de que

aquele que melhor se adapta aos critérios do processo de avaliação seja predominante, perpetuando assim o ciclo.

3.3.2. Características de um AG

Os AGs possuem um conjunto específico de características dentro desse universo dos algoritmos evolucionários os quais serão abordados ao longo do trabalho e necessitam aqui ser introduzidos.

Faz necessário reforçar o caráter probabilístico dos AGs em contrapartida a outros métodos determinísticos, que possuem soluções fixas para um determinado problema. Ao contrário desses métodos os AGs dada uma população inicial e um mesmo conjunto de parâmetro retornará uma solução diferente a cada vez que for executado sobre um determinado problema nessas condições.

Apesar do caráter probabilístico, os AGs possuem grandes vantagens em relação aos métodos convencionais determinísticos uma vez que são mais simples, necessitando apenas de informações inerentes ao ponto avaliado, sem que se faça necessário informações adicionais a respeito dos indivíduos do espaço amostral enquanto solução para o problema. Esse fator os torna extremamente aplicáveis a problemas reais que, de modo geral, incluem uma característica predominante de possuírem elementos discretos ou que não possuem derivadas.

Os AGs trabalham com uma grande quantidade de pontos, o que reforça sua aplicabilidade em casos reais, que muitas vezes, serão compostos de uma população muito grande de soluções. Contudo, isso também os diferencia dos métodos determinísticos uma vez que, sua região de busca irá se concentrar ao longo do processo em uma pequena parcela do espaço amostral, varrendo apenas um subconjunto do universo de soluções disponíveis. Essa característica é essencial, pois, como é o exemplo do PCV aqui utilizado como analogia do problema a ser tratado, o número de possíveis soluções é de caráter fatorial em função do número de elementos iniciais do problema, sendo, portanto, impossível a varredura por todo o espaço das soluções em um tempo considerado viável.

3.3.3 Elementos Básicos de um AG

Alguns elementos são comuns a todos os AG's, independente de sua aplicação, e para o esclarecimento de sua estrutura faz-se necessário que sejam definidos antes de que seja

explicado o funcionamento básico de um AG. Esses elementos são: os indivíduos, a avaliação (*fitness*) e os operadores de cruzamento.

Começando pelos indivíduos, são composições geradas pelo código que carregam informações do problema consigo e que representam para o algoritmo as possíveis soluções do problema. Esses indivíduos geralmente são compostos por cromossomos, que são a representação dos caracteres que compõem a solução, exemplificando com o caso do PCV, os cromossomos indicariam para quais cidades e em qual ordem o caixeiro deveria seguir. Dentro dos elementos conhecidos como indivíduos, além das informações trazidas pelos cromossomos, existem atributos que definem de forma mais completa as possíveis soluções para o problema como por exemplo: identificações, valores, indicadores, entre outros.

O elemento avaliação, conhecido como *fitness*, dentro dos AG's carrega uma importante função, que é a de caracterizar as possíveis soluções (indivíduos), baseados nas condições impostas pelo programador. Sua função é a de classificar as soluções como boas ou ruins em função de determinados critérios que podem ser, por exemplo, a seleção dentre os indivíduos do menor valor encontrado, como é o caso do PCV. Ele realiza essa classificação se utilizando de dados fornecidos por esses indivíduos e de seu método de avaliação imposto como critério, esse elemento será portanto a seletividade de todo o algoritmo, retornando ao final do processamento aquilo avaliado por ele como a melhor resposta ao problema.

Os operadores de cruzamento que também podem ser chamados de Crossover, são os responsáveis pelo cruzamento dos cromossomos de 2 indivíduos selecionados de modo a gerar descendentes e, através dessas sucessíveis combinações como ocorre na vida real, gerar soluções cada vez melhores do ponto de vista do que o problema exige. Diversos métodos e técnicas são utilizados para a construção desses operadores: cruzamento de dois pontos, cruzamento uniforme, onde cada um deles realiza a combinação de cromossomos de modo diferente o que produz diferentes alterações na amostragem de dados de um problema.

3.3.4. Processamento de um AG

Dentro das etapas constituintes de um AG a primeira etapa é realizada sobre um determinado grupo de indivíduos aqui chamados de população. O conceito de população é de fundamental importância para um AG e pode afetar grandemente a sua performance e seu objetivo, uma vez que, se trata da codificação da informação para o universo dos dados binários. Os indivíduos representam os elementos que compõe a base do problema, em um problema NP-completo como é o caso presente nesse trabalho, tais elementos são muito numerosos contudo

possuem certas características que compartilham entre si, e que, se forem codificadas eficientemente para o algoritmo conterão, além dos caracteres individuais dos elementos, caracteres inerentes ao próprio problema, como certas restrições, que acarretam em um incremento da eficiência da solução uma vez que serão economizados tempo e processamento. Essa economia é referente aos gastos que o algoritmo teria se tivesse que, por exemplo, avaliar se determinada solução encontrada é possível ou não, quando na verdade já poderia ter sido declarada que determinada característica, vinda de determinado indivíduo, não pode compor uma solução para o problema.

O ponto seguinte ao da codificação da população é a ação dos operadores, que se encarregaram de combinar as possíveis soluções para o problema, direcionados pelo caráter da função avaliação, com o objetivo de obter uma solução boa. É importante ressaltar que, dentro desse processo, o conceito de convergência genética ou perda da diversidade é de suma importância para a construção dos operadores enquanto algoritmo, e possui grande impacto sobre a qualidade da solução encontrada ao final das etapas.

3.3.5. Convergência Genética e Operador de Mutação

Segundo Linden (2012) a convergência genética é o caráter de uma população com baixa diversidade genética, que, por possuir genes similares, não consegue evoluir, a não ser pela ocorrência de mutações positivas. Ainda em Linden (2012) é descrito que a perda da diversidade pode ser definida como sendo o número de indivíduos que nunca são escolhidos pelo método de seleção de pais. Seriam esses elementos do espaço amostral aqueles que recebem uma avaliação baixa e por isso tendem a não serem selecionados pelos operadores genéticos para a confecção de um novo indivíduo.

Tais características são negativas para busca de soluções boas para o problema, uma vez que impedem a evolução das soluções encontradas pelo algoritmo. Esses fatores irão produzir uma restrição aos operadores e a função avaliação, que são os responsáveis pela varredura e operação em todo espaço amostral, fazendo com que o algoritmo convirja à uma determinada espécie de solução.

Na tentativa de contornar esse impasse é que são utilizados os operadores de mutação. Esses elementos são os responsáveis por modular, em suas limitações, as mutações que ocorrem na natureza ocasionalmente e que são responsáveis por prover características que tornam certos indivíduos mais aptos a sobreviver do que outros. Esses operadores, assim como os

responsáveis pelos cruzamentos, possuem diversas técnicas para sua construção, que tornam variável o seu comportamento e sua influência dentro do AG.

3.4. COMENTÁRIOS

Neste capítulo foi apresentado a importância do estudo dos Algoritmos Genéticos, bem como a origem da base da técnica utilizada nesse trabalho: Algoritmos Evolucionários. Posteriormente, foram abordadas a Definição computacional do AG, as Características de um AG, os Elementos Básicos de um AG, o Processamento de um AG e por fim, a Convergência Genética e Operador de Mutação.

4. METODOLOGIA

Nesse capítulo serão apresentados todos os processos, de um modo mais abrangente, que levarão à confecção da solução do problema proposto. A pesquisa tem início pela elaboração da parte introdutória do trabalho que ficará responsável por apresentar os conceitos necessários para o entendimento do problema bem como sua solução. Após isso, são apresentados alguns conceitos absoldidos do levantamento bibliográfico necessário para a estruturação e fundamentação da pesquisa e, finalmente, a metodologia usada na aplicação dos conhecimentos obtidos sobre o problema proposto como forma de solução para o problema.

4.1. INTRODUÇÃO AO PROBLEMA E A FERRAMENTA

Inicialmente será apresentada a conceituação dos problemas NP-Completo, assim como a descrição do problema a ser tratado e sua modelagem. Essa parte inicial traz consigo a missão de desvendar ao leitor quaisquer dúvidas a respeito das associações que serão feitas entre o problema proposto, que possui caráter de um problema real da logística atual, e seus semelhantes no acervo teórico a respeito dos problemas que tratam de rotas e combinações de possibilidades.

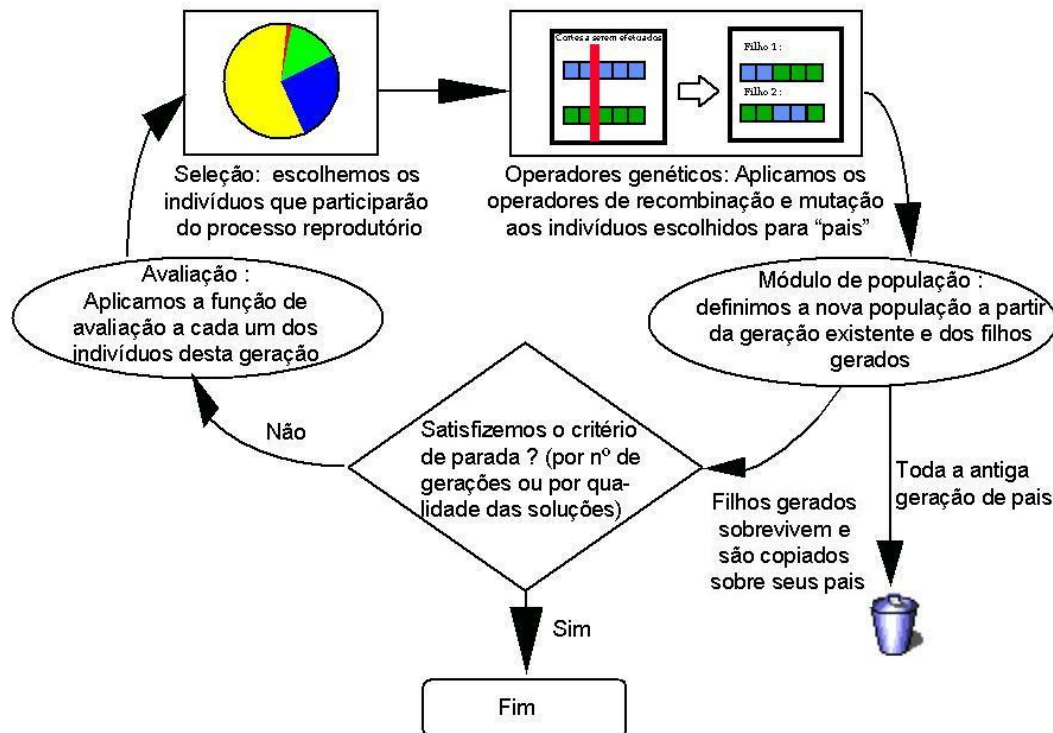
A apresentação dos conceitos que envolvem os problemas NP-completos e o PM se faz necessária pois o problema a ser tratado, utilizando o método dos AGs, é análogo a esses tópicos que foram referenciados em teoria, sendo que alguns parâmetros e condições serão mudados no caso real a ser avaliado ao longo desse trabalho.

Um organograma básico da ferramenta aqui utilizada (AG), é mostrado na Figura 2, para elucidar o processo e a estrutura que irá compor todo o corpo do algoritmo criado ao longo do trabalho. Essa mesma estrutura será replicada para a construção do AG ao final do trabalho, sendo que esses conceitos irão se traduzir em determinadas operações dentro da linguagem de programação escolhida.

A linguagem de programação escolhida para a criação da ferramenta é o Python 3.7, que é a versão mais recente da linguagem até a data de realização desse trabalho. Tal escolha foi fortemente motivada pelas facilidades que essa linguagem de programação possui e que vem ganhando muito espaço na comunidade acadêmica, oferecendo: agilidade no processamento do código (o que virá a ser explicado mais a frente quando o Algoritmo final for apresentado), linguagem simples de comandos e funções por se tratar de uma linguagem de programação de

altíssimo nível (o que facilita o trabalho com a linguagem independentemente do nível de conhecimento e experiência do operador a respeito da mesma) além é claro, de sua fonte livre e aberta e de seu rico conteúdo disponível, que abrange desde desenvolvimento web até inteligência artificial.

Figura 2 - Esquema de um algoritmo genético.



Fonte: LINDEN (2012).

4.2. O PROBLEMA A SER TRATADO

É importante que seja esclarecida a natureza e conceituação do problema a ser aplicado mais a frente, apesar de sua completa analogia com o PM. O objetivo desse trabalho é solucionar um problema de logística, criado unicamente para aplicação neste trabalho, que possui a seguinte descrição: um motorista deseja alocar uma determinada carga de produtos, onde cada produto possui um determinado valor em reais (R\$) e volume em metros cúbicos (m³), em um caminhão de volume livre total de L metros cúbicos. O objetivo do motorista é alocar uma carga de maior valor possível dentro do caminhão, utilizando o menor espaço que conseguir, maximizando assim sua eficiência no carregamento.

O problema exige que os dados relativos aos produtos, valores e volumes, sejam informados pelo operador, assim como seja informada a capacidade total do caminhão. Esses

dados serão cruciais para o algoritmo uma vez que serão esses os critérios que o mesmo utilizará para definir a melhor solução para o problema. Serão feitas duas variações distintas utilizando esse caso, sendo que, em um dos testes a carga disponível para ser locada dentro do caminhão conterà apenas uma unidade de cada produto disponível, e em um segundo momento, serão disponibilizadas diferentes quantidades de cada produto para serem trabalhadas.

O objetivo do AG é que determine a melhor solução possível para o carregamento do caminhão de volume L declarado, com os produtos que forem informados pelo operador. Ao final serão exibidos: a melhor solução para o problema, assim como sua geração, produtos que foram selecionados, preço, espaço ocupado e um mapeamento gráfico do desenvolvimento das melhores soluções ao longo das gerações percorridas pelo algoritmo.

4.3 MODELAGEM

Modelar qualquer situação ou problema do mundo real para a linguagem utilizada pelas máquinas computacionais exige do programador um pensamento a frente do problema, de forma que, todo o problema terá uma estrutura onde suas partes são interconectadas do início ao fim. Para tanto é necessário em um primeiro momento que seja feito um planejamento do “esqueleto” de todo o problema transferindo suas características para a lógica de programação.

A base utilizada para a montagem desse “esqueleto” é o esquemático mostrado na Figura 2. O algoritmo resultante desta modelagem, como mostrado na Figura 3, passa por um critério de satisfação que, enquanto não for correspondido, irá avaliar cada indivíduo da população gerada, selecioná-los para o cruzamento, aplicar os operadores genéticos de cruzamento e mutação e em seguida irá descartar a antiga população substituindo-a pelos descendentes da anterior. Todo esse processo lógico se repete enquanto o critério de parada não for satisfeito, sendo esses parâmetros de paradas definidos pelo usuário.

Todas essas ordenações de ações provem da lógica de programação abstraída dos processos evolutivos advindos da biologia. Cada um dos passos representa uma etapa análoga as que são encontradas na natureza, e todos eles se integram para formar um sistema que, com base na avaliação dos indivíduos presentes na população irá realizar manipulações de forma contínua, com o intuito de evoluir cada vez mais a “espécie” até corresponder os critérios de parada.

Figura 3 – Algoritmo Genético Abstrato.

```

# Algoritmo Genético Abstrato
1 Recebe os critérios de Parada
2   Inicializa População
3   Avaliar População
4   while Critérios de Parada não satisfeitos:
5       Seleciona os Pais
6       Aplica o operador de cruzamento
7       Aplica o operador de mutação
8       Avalia a população
9       Descarta a população antiga
10      Defina a população nova
11
12  # Uma vez satisfeito o critério de parada
13  Lista os melhores indivíduos
14
15  Fim do Algoritmo
16

```

Fonte: Acervo Próprio (2019).

A modelagem pode, portanto, ser subdividida entre os diferentes componentes do algoritmo que, apesar de necessitarem interagir entre si para realizarem as tarefas exigidas pelo processo, são funções independentes e podem ser alteradas conforme se mostre necessário para o problema.

4.3.1 Modelagem do Indivíduo

A principal base para a confecção do algoritmo aqui presente é o que na programação é chamado de POO (Programação Orientada a Objeto), o intuito dessa técnica é justamente modelar coisas do mundo real para linguagem de computadores. O uso de POO para a execução dessas tarefas consiste em atribuir a uma determinada classe (que representa o objeto a ser modelado) os atributos necessários para que o algoritmo possa compreender aquilo que tal classe pode armazenar, executar, atribuir como demonstrado na Figura 4. Nela está representada a classe com seus devidos atributos, que serão utilizados e manipulados durante a execução do AG por outras classes ou funções.

Como representa o primeiro elemento do problema, a modelagem teve início pela representação de um caminhão fictício e de suas possíveis cargas que são os responsáveis por representar: o alpinista e seus itens da mochila respectivamente. Para poder representar esse conjunto faz necessário primeiro a modelagem dos itens para o algoritmo, representados pela classe Produto mostrada no algoritmo abstrato da Figura 4.

Figura 4 - Definição da classe Produto (Abstração)

```

class Produto():
    recebe os atributos (nome, espaço, valor):
        armazena em x = nome
        armazena em y = espaço
        armazena em z = valor

```

Fonte: Acervo Próprio (2019).

Como o PM descreve, cada item que pode ser alocado dentro da mochila, que para o caso montado é representada pelo caminhão, recebe como característica um valor e um volume que serão os fatores que mais a frente o algoritmo irá utilizar para avaliar cada um dos produtos e sua inclusão ou não dentro do caminhão. Seguindo estas exigências, assim como na Figura 4, a classe ou objeto criado no algoritmo que representa estes itens deverá receber essas atribuições.

Após a definição dos objetos que comportaram os dados que compõem o problema (Produtos), é necessário que seja modelado o indivíduo, que será a exata representação das possibilidades de solução para o problema, aplicado a cada uma das situações abordadas.

Para o caso criado, da alocação de diversos produtos em um espaço limitado (Caminhão), a modelagem dos indivíduos (soluções) se resume em uma distribuição binária, que possui tantos elementos quanto produtos disponíveis, sendo atribuído a cada produto um número nessa distribuição, e que representa, de modo literal, se um determinado produto será levado ou não. Para essa distribuição de “zeros” e “uns”, os produtos aos quais o número “0” for atribuído não serão levados no caminhão e aqueles que receberem o número “1” serão levados, cada arranjo desses compõem uma solução (indivíduo) que dirá quais produtos o “alpinista” irá levar. Segue na Tabela 1 a representação de um exemplo de indivíduo, onde tem-se uma amostragem com 8 produtos disponíveis, representados por letras de A à H, onde os produtos que recebem o valor “0” não serão levados e os que recebem o valor “1” serão os escolhidos como carga para o caminhão fictício.

Tabela 1 - Definição do Indivíduo

| Indivíduo 1 | A | B | C | D | E | F | G | H |
|--------------------|----------|----------|----------|----------|----------|----------|----------|----------|
| | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |

Fonte: Acervo Próprio (2019).

O arquétipo de cada indivíduo ou solução terá o formato semelhante ao do visto na representação da Tabela 1 uma vez que, dada uma certa quantidade de itens, o padrão das soluções manterá o padrão de sequências binárias, que fazem analogia ao “cromossomo” do indivíduo modelado para o problema. Esta associação auxilia na modelagem do indivíduo dentro do algoritmo assim como sua manipulação pelos operadores de cruzamento e mutação.

Dentro do algoritmo, para fins de processamento, o objeto criado para definir o indivíduo deve receber certos parâmetros que permitam ao algoritmo reconhecer e operar os valores de cada produto afim de encontrar a melhor resposta.

A classe definida na Figura 5 traz consigo todas as informações das quais o AG necessitará do indivíduo que for formado através dela, cada umas das características mostradas são definidas com o intuito de modelarem de maneira completa a carga que será levada em um caminhão de capacidade a ser definida pelo usuário.

Figura 5 - Definição da classe Indivíduo (Abstração)

```
class Indivíduo():
    recebe os atributos (espaços, valores, limite_espacos,
                        nota_avaliacao, espaco_usado,
                        geração):
        armazena em a1 = espaços
        armazena em a2 = valores
        armazena em a3 = limite_espacos
        armazena em a4 = geração
        nota_avaliacao = atribui ao individuo
        espaco_usado = atribui ao individuo
        cromossomo = matriz linha
```

Fonte: Acervo Próprio (2019).

Esse objeto é composto por algumas informações que são necessárias para que o AG realize a avaliação do indivíduo gerado, a verificação da possibilidade desse indivíduo ser uma resposta viável, além é claro, poder realizar operações como cruzamento e mutação desse indivíduo com outros gerados para assim encontrar a melhor resposta possível. Faz-se necessária, portanto uma elucidação do significado e função dessas partes que compõe o indivíduo modelado.

Dentro do escopo dessas informações das quais esse objeto recebe está a variável “espaços”. Essa variável nada mais é do que aquela responsável por armazenar o espaço ocupado por cada um dos produtos presentes na solução representada pelo indivíduo. Para um determinado arranjo de produtos, aqueles que receberem o valor “1” terão seus valores, relativo

ao espaço que ocupam, retirados dessa variável, que é extraída do banco de dados que o usuário cria e atribui ao objeto “Produtos”.

O próximo atributo presente é a variável “valores”, que irá realizar o mesmo armazenamento da variável anterior, com a única diferença que serão trabalhados os valores de cada um dos produtos, os mesmos serão extraídos da mesma forma que a variável anterior da classe “Produtos”, esses valores serão computados juntamente da variável “espaços” para que cada indivíduo seja avaliado enquanto solução para o problema, tomando como referência as condições estabelecidas no método “avaliação” que será explicitado mais à frente.

A variável seguinte “limite_espacos” é um limitador definido pelo usuário e que é uma característica atribuída sobre cada indivíduo, criada para que o mesmo não extrapole o limite de carga que o “caminhão” consegue suportar, esse valor é um dos atributos responsáveis pela avaliação da viabilidade de cada solução gerada ser uma resposta viável para o problema ou não.

Os atributos seguintes “nota_avaliacao” e “espaço_usado” são utilizados para computar dentro da função “avaliação”, para que sejam armazenados os valores que representam, respectivamente, o somatório dos valores da carga e o somatório do espaço ocupado por ela. Esses atributos serão, portanto, as variáveis que armazenam os valores que representam o *fitness* dado a cada uma das soluções geradas.

Seguindo na ordem mostrada na Figura 5, o atributo “geracao” é utilizado para que cada indivíduo tenha uma contagem atribuída a essa variável, que indica qual a numeração da geração determinada a solução pertence. Ao longo do processamento do AG a contagem da geração de cada indivíduo será importante para avaliar em que momento o algoritmo encontrou a solução que ele retorna como sendo a mais efetiva, possibilitando que sejam feitas alterações na tentativa de se obter cada vez melhores soluções. Essa avaliação feita com base na observação do número da geração em que a melhor solução foi obtida, permite observar a variação na amostragem dos resultados, quando mudados certos parâmetros da ferramenta como tamanho da população, número de gerações a serem computadas, entre outros fatores.

Por fim o objeto indivíduo tem como atributo sua composição primordial que é o cromossomo, representado anteriormente por uma matriz de linha única e diversas colunas, com elementos zeros e uns, que traduzem ao algoritmo que produtos estão sendo levados ou não. O termo cromossomo advindo da biologia traduz de maneira análoga o que esse elemento representa para o AG, que seriam as próprias características expressas, inerentes unicamente a aquele indivíduo, e que podem ser combinadas com outros, ou sofrerem mutação gerando descendentes.

4.3.2 Modelagem da Avaliação (*Fitness*)

Avaliar o indivíduo enquanto solução para o problema é a etapa de controle de critérios do algoritmo, pois, é através desse procedimento que o AG conseguirá selecionar os melhores, indivíduos no intuito de fazer atuar sobre eles os operadores de cruzamento, na tentativa de encontrar resultados ainda melhores.

A avaliação do AG neste trabalho foi confeccionada com base no método elitista, que define as melhores soluções com base no seu parâmetro “nota_avaliacao”, ordenando as soluções da que tem a maior “nota_avaliacao” para as que tem a menor. O elitismo sempre colocará como melhor solução do problema, portanto, o indivíduo da população que tiver o maior somatório do valor de utilidade e corresponder ao critério de espaço ocupado.

A abstração do função avaliação, mostrada na Figura 6, responsável pela avaliação dentro do algoritmo realiza o somatório dos valores utilidade dos itens do Indivíduo, o somatório dos espaços que eles ocupam, e ao final irá verificar se ele corresponde a uma solução viável para o problema através da condição imposta pelo limite de espaço do caminhão, caso corresponda a uma solução viável lhe é atribuída como nota o somatório do valor de utilidade dos itens que selecionou levar na “mochila”, caso contrário o indivíduo recebe uma nota de valor igual a 1.

Figura 6 – Função Avaliação (Abstração)

```

1 função Avaliação():
2     if gene no cromossomo do Indivíduo = 1
3         nota de avaliacao = soma dos valores
4         de utilidade dos itens
5         espaço usado = soma dos espaços ocupados
6         pelos itens
7
8     nota do individuo = nota de avaliacao
9     espaço do individuo = espaço usado

```

Fonte: Acervo Próprio (2019).

Para que as soluções inviáveis não tenham valor como boas soluções dentro do espaço amostral, sempre que um Indivíduo extrapola o limite de espaço definido pelo limite disponível do caminhão essa solução recebe uma nota baixa, definida para este trabalho como sendo de

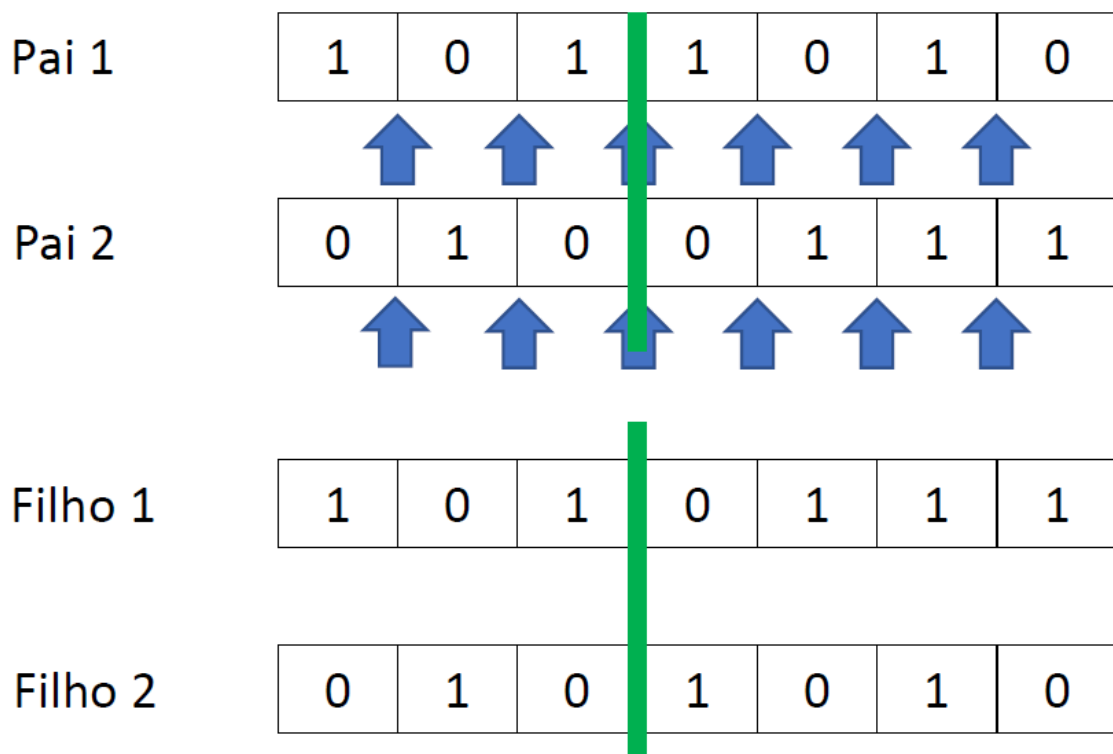
valor unitário, esse indivíduo portanto, não ficará excluído da população, mas terá mínimas chances de ser selecionado para gerar descendentes.

4.3.3 Modelagem do Operador de Cruzamento

O operador de cruzamento será o método do AG responsável por ordenar o cruzamento dos cromossomos dos dois Indivíduos selecionados como pais. O cruzamento de dois “pais” tende a gerar descendentes caracterizados como soluções mais aptas e, com o avanço das gerações, a população tende a evoluir em qualidade de solução.

Foi definida para este trabalho a utilização do método de cruzamento de um ponto, onde, considerando o cromossomo de cada Indivíduo, um ponto aleatório é selecionado para que o cruzamento ocorra de forma simétrica alternada, conforme mostra a Figura 7.

Figura 7 - Cruzamento de um ponto.



Fonte: Jones Granatyr (2018).

O operador executa, portanto, após ter selecionado um ponto aleatoriamente definido na cadeia dos cromossomos, a troca “genética” entre os genes dos pais para gerar os descendentes. De forma direta, como mostra a Figura 7, o Filho 1 recebe os três primeiros genes do Pai 1 e os

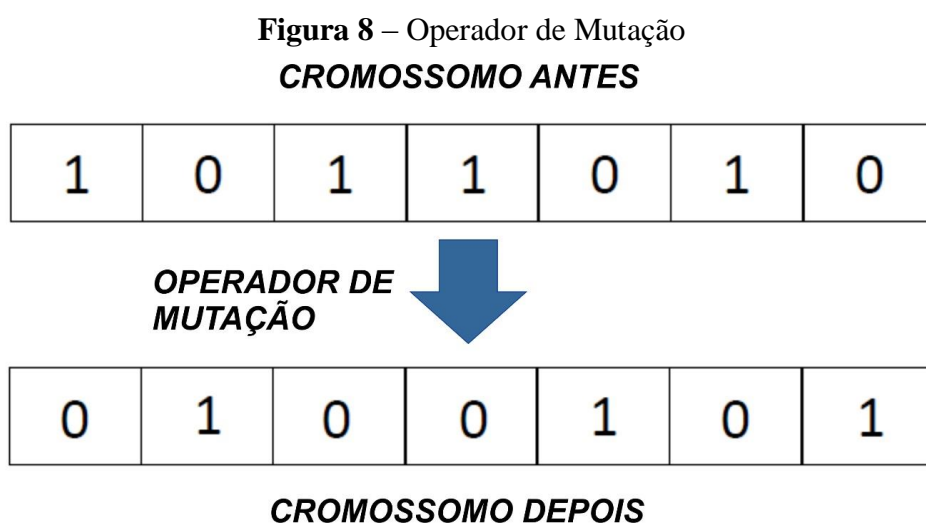
genes restantes do Pai 2. Em seguida, na formação do Filho 2, ele recebe os três primeiros genes do Pai 2 e os genes restantes do Pai 1.

4.4.4 Modelagem do Operador de Mutação

Modelar o que seria o processo de mutação que ocorre na natureza é algo bem distinto no campo dos algoritmos, uma vez que, esse método é realizado sobre os cromossomos do Indivíduo para o caso em questão. Quaisquer alterações de genes, sendo elas aleatórias que seriam o caso que mais se aproximaria daquilo que acontece no mundo tangível, afetam grandemente as soluções representadas pelo Indivíduos.

As mutações nos AGs criam variações, através de alterações randomizadas, que alteram os genes ou partes menores do cromossomo de forma a gerar variações, que, tem o intuito de fazer a manutenção da diversidade genética na população. Pode ser feita uma analogia mais técnica em relação aos operadores de mutação, como sendo agentes que injetam novos cromossomos na população permitindo ao algoritmo buscar soluções fora dos limites definidos pela população inicial (Linden, 2012).

O processo mostrado na Figura 8, ilustra de forma literal o processo que ocorre dentro do operador de mutação utilizado para compor o AG deste trabalho. O operador age sobre os genes do cromossomo invertendo-os, portanto, para os genes de valor “0” terão seu valor alterado para “1” e vice-versa.



Fonte: Acervo Próprio (2019).

Esse operador está atrelado a uma taxa de ocorrência definida pelo usuário, essa taxa irá definir a frequência de ação desse operador sobre os indivíduos da população, mais especificamente sobre os filhos de uma determinada geração que será quando os operadores de mutação entrarem em ação. A taxa de ocorrência, na maior parte dos trabalhos na área de AGs usa um valor pré-determinado de 0,5% ou 1%, influenciados pelos bons resultados obtidos em trabalhos anteriores que se utilizaram dessa mesma taxa (Linden, 2012). Em trabalhos anteriores citados por Michalewicz (1996) uma ótima taxa de mutação para problemas de otimização de cromossomos binários, como é o caso aqui trabalhado, é igual a $1/L$, onde L é igual ao número de variáveis binárias.

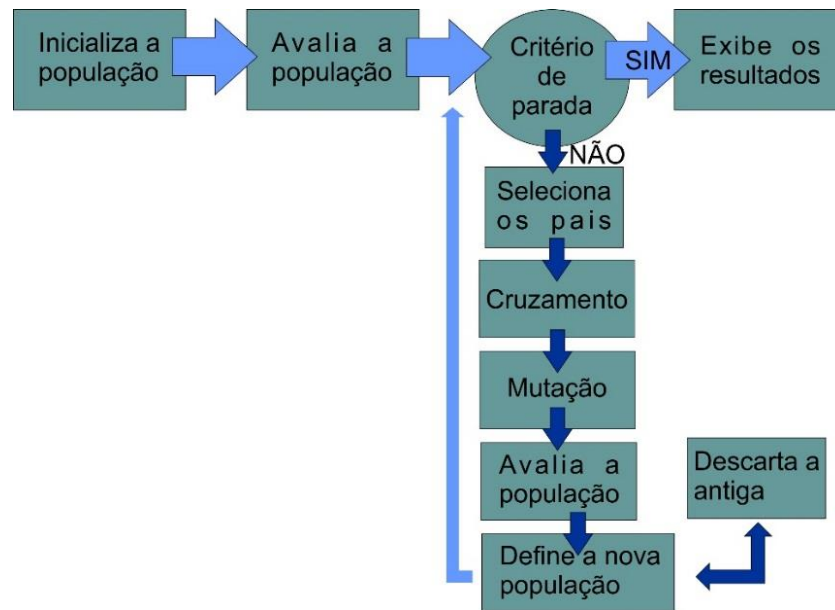
Estes valores comentados costumam oferecer resultados interessantes para as aplicações, contudo, faz-se necessário lembrar que o ideal é que sejam feitas tentativas para encontrar a melhor taxa de mutação para o problema que está sendo trabalhado, pois algo direcionado terá maiores chances de eficácia do que um padrão pré-definido.

4.4 CONSTRUÇÃO DO ALGORITMO

Intuitivamente o algoritmo necessita unir todos os elementos até então abordados para que, através de um processo contínuo, seja capaz de buscar a melhor solução para o problema, dentro do espaço amostral gerado, utilizando os métodos já abordados nesse trabalho. Experimentalmente o algoritmo foi organizado em uma determinada ordem de ação que faz alusão ao ciclo exibido na Figura 2.

O AG será formado por um algoritmo que utiliza todos os elementos modelados anteriormente, postos em um determinado loop (laço condicional) que terá como condição de parada o número de gerações determinado pelo usuário. Essencialmente o algoritmo deve gerar uma população de soluções e, após isso, avaliar estas soluções para depois aplicar os operadores sobre essa população no intuito de direcioná-la para a melhor otimização possível, repetindo o ciclo até a condição de parada.

O loop executado pelo AG está representado na Figura 9, seguindo a sequência de processos anteriormente comentada. Vale ressaltar o descarte da antiga população sempre que uma nova população é criada através da atuação dos operadores genéticos (operador de cruzamento e mutação). Para essa nova etapa do processo surge uma nova função que deve ser realizada pelo AG que será abordada logo mais, o método responsável por selecionar os pais.

Figura 9 – Organograma do AG montado

Fonte: Acervo Próprio (2019).

4.4.1 Método “Seleciona_pais”

O método em questão é a função dentro do algoritmo aqui utilizado responsável pela seleção dos geradores, indivíduos dentro da população, que serão aqueles cuja descendência irá substituir a antiga, sobre a pretensão de serem soluções melhores para o problema que seus antecessores. Apesar do intuito do algoritmo ser sempre a busca por soluções melhores, como existem processos randomizados dentro do algoritmo, que tem o intuito de trazer maior diversidade combinatória para o método, nem sempre os descendentes destes geradores serão soluções mais notáveis para o problema, contudo, o AG deve ser capaz de utilizar esses indivíduos para gerar soluções ainda melhores para o problema com o passar do seu processamento.

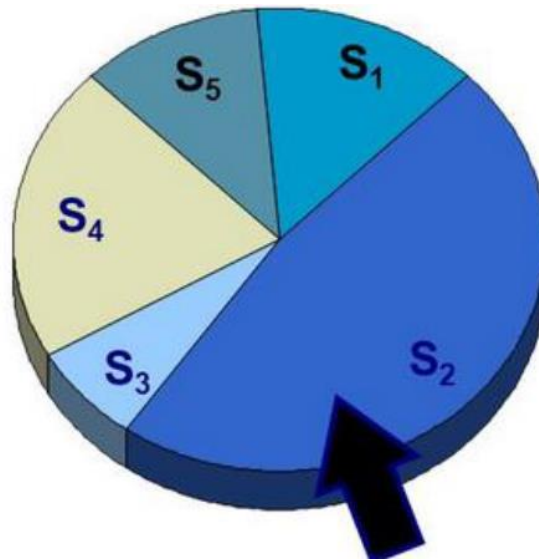
A base para a construção do método que seleciona os pais será o da roleta viciada, que consiste na seleção dos indivíduos com base na sua nota de avaliação. Segundo Linden (2012), “Quando montamos uma roleta viciada para uma determinada população, somamos todas as avaliações e para cada individuo alocamos um espaço igual à avaliação deste individuo dividida pela soma das avaliações de todos os indivíduos.”

Essa metodologia tem por objetivo reproduzir o método de seleção natural que atua sobre as espécies encontradas na natureza, onde pais mais aptos geram descendentes mais frequentemente do que pais menos aptos, embora esses últimos também venham a gerar filhos.

Trazendo a analogia para dentro do algoritmo, indivíduos com maior nota de avaliação serão selecionados como pais mais frequentemente que indivíduos com nota de avaliação menor, fazendo com que a característica dos melhores indivíduos predomine na nova população formada pelos descendentes. Deste modo, os indivíduos com maior nota de avaliação recebem privilégios, contudo, aqueles com avaliação menor não são desprezados. O intuito dessa manutenção da permanência de características dos indivíduos com nota de avaliação menor é, novamente, a conservação de um certo grau de diversidade genética dentro da população, para que soluções ainda melhores possam vir a serem encontradas.

A Figura 10 exibida, ilustra a roleta-viciada criada, selecionando cinco indivíduos da população, e utilizando como medida para calcular seu espaço ocupado na roleta sua nota de avaliação. No exemplo mostrado, o indivíduo S_2 possui a maior nota de avaliação dentre os 5 e por isso ocupa um espaço maior dentro da roleta. Esse fator fará com que ele seja selecionado mais facilmente do que o indivíduo S_3 que ocupa um espaço bem menor dentro da roleta, contudo não exclui esse último de poder ser selecionado como um indivíduo para a formação de descendentes da nova população.

Figura 10 – Ilustração do método da roleta-viciada.



Fonte: Jones Granatyr (2018).

Essa proposta, assim como na avaliação do indivíduo traz consigo o caráter do elitismo que, utilizando certos métodos, atribui vantagens aos indivíduos mais bem avaliados, na intenção de que suas características venham a perdurar nas diversas populações que serão geradas.

4.4.2 Abstração do AG

O código montado em linguagem Python, como abordado anteriormente, utiliza a metodologia de POO para sua construção. Sendo criada uma classe denominada “AlgoritmoGenetico” que carregará consigo todas as funções necessárias para a execução da lógica demonstrada na Figura 9. O intuito é que o algoritmo realize as seguintes etapas quando for executado pelo usuário:

- 1° - Inicialize a população.
- 2° - Ordene a população pela nota de avaliação (em ordem decrescente).
- 3° - Defina o melhor Indivíduo na geração atual.
- 4° - Realize a soma das notas de avaliação dos indivíduos da população.
- 5° - Selecione os pais através do método da roleta viciada.
- 6° - Realize o processo de cruzamento através dos operadores de cruzamento.
- 7° - Caso for determinado, realize o processo de mutação através do operador.
- 8° - Substitua a antiga população pela nova.
- 9° - Avalie os novos Indivíduos da população.
- 10° - Ordene a população com base no método elitista.
- 11° - Aloque o melhor Indivíduo na primeira posição da população.
- 12° - Exiba a geração atual informando: o melhor Indivíduo, seu valor de utilidade total e seu espaço total ocupado.

O algoritmo segue em sua execução esta sequência lógica, mostrada pelos passos de 1 a 11, para que ao final seja possível visualizar a melhor solução encontrada por ele a cada nova geração. Ou seja, a cada nova ação dos operadores de cruzamento e mutação, que são os motores combinatórios do AG, é possível visualizar a melhor solução por eles encontrada, através desses resultados será exibido um gráfico em função do avanço da contagem da população que servirá como avaliação do desempenho do algoritmo ao longo de seu funcionamento.

Na Figura 11 é possível visualizar a união de todos os métodos até então citados, organizado nessa cadeia de processos, que resulta no conjunto principal da ferramenta de solução do problema deste trabalho. A abstração do AG mostrada tem por intuito esclarecer o ciclo de etapas percorrido pelo algoritmo na busca pela solução do problema.

O usuário define o dado “tamanho da população” recebido pelo AG, que o armazena para que seja utilizado dentro da função que gera a população inicial, a variável irá definir o

número de indivíduos que as populações irão ter ao longo do processamento do AG. Com a população inicial gerada é feita a ordenação para que os melhores indivíduos, mesmo que não correspondam a soluções viáveis, para que então possa ser feito o somatório da avaliação de cada um dos indivíduos.

A etapa seguinte do algoritmo consiste na aplicação das metodologias até então citadas, onde ocorrerá a aplicação do método da roleta viciada para seleção dos pais, a ação dos operadores de cruzamento para realizar a combinação desses pais, e, caso for determinado pela taxa de mutação, ocorrerá a ação do operador de mutação sobre os descendentes gerados. Esse processo de geração de uma nova população é seguido da sua ordenação e avaliação com base no elitismo seguido de sua exibição na tela do usuário, finalizando assim uma geração do algoritmo. Lembrando que o ciclo de execução perdura enquanto o algoritmo não corresponder a condição de parada, que é representada pelo número de gerações determinados pelo usuário.

Figura 11 – Escopo do AG utilizado (Abstração)

```

1 class AlgoritmoGenetico():
2     recebe o atributo ( tamanho da população ):
3     armazena em x = tamanho da população;
4
5     Executa a função Inicializa_a_população;
6
7     Executa a função Ordena_a_população;
8
9     Executa a soma das notas de avaliação;
10
11    Executa a função Selecciona_pais;
12
13    Executa o método Crossover(Cruzamento);
14
15    if correspondida a condição:
16        Executa o método Mutação sobre os
17        descendentes gerados;
18
19    Executa a função Ordena_a_população;
20
21    Aloca o melhor Indivíduo na primeira
22    posição da nova_populacao;
23
24    Exibe a Geração atual, o melhor Indivíduo
25    seu valor de utilidade e espaço usado totais;
26
27    if correspondido o critério de parada
28        Fim da execução

```

Fonte: Acervo Próprio (2019).

4.3. COMENTÁRIOS

Neste capítulo foi apresentada a metodologia proposta por esse trabalho. Esta metodologia consiste inicialmente na introdução ao problema e a ferramenta, bem como o problema a ser tratado e sua modelagem. Também foi mostrado como foi realizada a modelagem do indivíduo, modelagem da avaliação (*fitness*), modelagem do operador de cruzamento e a modelagem do operador de mutação. Desta forma foi possível demonstrar a construção do algoritmo assim como o método “seleciona_pais” e a abstração do ag.

5. RESULTADOS E DISCUSSÕES

5.1. INTRODUÇÃO

A utilização de AG como ferramenta de solução para o problema proposto, devido a sua natureza estocástica derivada das combinações aleatórias feitas pelo algoritmo no intuito de encontrar a melhor resposta, apresenta em repetidas execuções resultados diferentes que podem ou não, vir a serem soluções consideradas ótimas para o problema.

Devido a esta característica os resultados foram avaliados conforme seus valores de utilidade e espaço ocupado, e o intuito dos experimentos realizados com o algoritmo foi o de aproximar cada vez mais esses valores das melhores condições possíveis. Portanto, o objetivo é de que os valores de utilidade sejam maximizados e o espaço ocupado pela melhor solução encontrada seja o mais otimizado possível.

A máquina sobre a qual foram feitas as experimentações com o AG possui as seguintes características técnicas:

Processador: Intel® Core™ i3-2100 CPU @ 3.10GHz 3.10GHz

Memória (RAM): 4,00 GB

Tipo de sistema: Sistema Operacional de 64bits

Versão Python: 3.7 64-bits

5.2. ELABORAÇÃO DOS TESTES

As condições impostas sobre o algoritmo que estão passíveis à definição do usuário, como, tamanho da população, limite de espaço suportado pelo caminhão, número de gerações, taxa de mutação, e é claro, as características dos itens com as quais o algoritmo terá que trabalhar foram definidas com base na natureza dos experimentos feitos. O intuito é adequar da melhor maneira possível as características do algoritmo em função da natureza do problema que está sendo tratado.

Como comprovação da eficácia do método aqui utilizado o intuito é que inicialmente sejam feitas aplicações do AG construído sobre conjuntos de dados disponíveis online e que são regulamentados pela *GNU LGPL license*. Essa licença garante o compartilhamento de

bancos de dados que são de livre acesso e uso pelo usuário, para aplicação em diversos problemas. Essas aplicações variam de problemas relacionados a análises combinatórias, indo até os que envolvem problemas NP-difíceis e NP-completos. Diversos conjuntos de dados disponíveis estão relacionados ao PM e possuem variantes quanto a natureza do problema e as ferramentas utilizadas para solucioná-lo, assim como também trazem consigo a solução ideal para o caso que descrevem, obtida através de outros métodos como o de força bruta por exemplo, que retornam de maneira assertiva a melhor solução possível. O intuito da utilização de um desses bancos de dados é aferir a eficiência do algoritmo desenvolvido enquanto atuando sobre um dos casos escolhidos, para que, logo mais o algoritmo possa ser aplicado em situações criadas e adaptadas a casos mais reais da logística que algumas empresas de transporte.

5.3. TESTE 1 – Problema P01

Como primeira experimentação do algoritmo o banco de dados escolhido foi o P01, presente no *Data for the 01 Knapsack Problem*, essa seção contém conjuntos de dados específicos do problema da mochila 0-1, que traz consigo as seguintes características:

Tabela 2 – Descrição do Problema P01

| PROBLEMA P01 | | |
|--------------------------|-----------------------|---------------------------|
| Limite da mochila | | 165 |
| Item | Espaço ocupado | Valor de utilidade |
| 1 | 23 | 92 |
| 2 | 31 | 57 |
| 3 | 29 | 49 |
| 4 | 44 | 68 |
| 5 | 53 | 60 |
| 6 | 38 | 43 |
| 7 | 63 | 67 |
| 8 | 85 | 84 |
| 9 | 89 | 87 |
| 10 | 82 | 72 |

Fonte: Adaptado de Data for the 01 Knapsack Problem (2014).

A amostragem de dados mostrada, simula uma determinada amostragem de itens, que necessitam serem arranjados em uma mochila que possui espaço limite de valor 165. A natureza das medidas envolvidas no problema não necessariamente precisa ser definida uma vez que os valores podem ser entendidos como montantes de mesma natureza, ou seja, entende-se que o valor Limite da mochila terá mesma unidade de medida que o peso ou espaço ocupado pelos itens, somado a isso os valores de utilidade de cada item indicam apenas a atribuição que cada um adiciona ao ser levado na “mochila”, ficando de livre interpretação ao usuário sua natureza, podendo ser por exemplo o valor monetário dos itens.

A melhor solução ideal definida pelo banco de dados para esse problema possui a seguinte forma:

Tabela 3 – Parâmetros utilizados para P01

| SOLUÇÃO IDEAL PROBLEMA P01 | |
|-----------------------------------|----------------------|
| Cromossomo | [1 1 11 0 1 0 0 0 0] |
| Peso Total/Espaço ocupado | 165 |
| Valor de utilidade | 309 |

Fonte Adaptado de Data for the 01 Knapsack Problem (2014).

Por meio da informação do banco de dados, é descrita como a melhor solução para o problema P01 a locação dos itens 1, 2, 3, 4 e 6 na mochila. Essa solução provém o maior valor de utilidade para a carga no peso ou espaço utilizado mais otimizado possível. O intuito é que o algoritmo construído seja, portanto, capaz de chegar até essa solução ou se aproximar com pequena variação dos valores descritos.

Os parâmetros usados inicialmente para execução do algoritmo sobre o problema foram:

| |
|---------------------------------------|
| Tamanho da População = 20 |
| TESTE 1 Taxa de mutação = 0,01 |
| Número de Gerações = 100 |

Utilizando esses parâmetros para o processamento do algoritmo sobre o problema P01 o AG, após repetir a mesma execução por um baixo número de tentativas, consegue retornar a melhor solução para o problema de acordo com o banco de dados.

Os resultados retornados pelo algoritmo estão representados nas Figuras 12 e 13. Na primeira o programa, após sua execução, retorna a melhor solução por ele obtida, informando

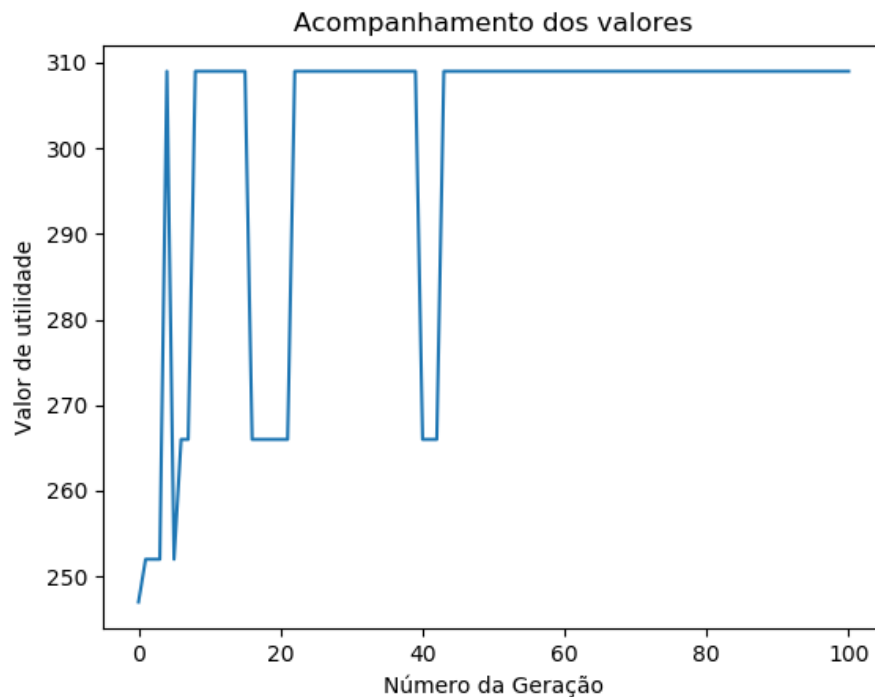
o número da geração em que foi encontrada, o valor de utilidade total desse indivíduo, o espaço que ele ocupa, a representação do seu cromossomo, logo abaixo a descrição dos itens escolhidos por essa solução e o tempo de duração do processamento do AG para que encontrasse esta resposta.

Figura 12 – Solução para o caso P01 encontrada pelo AG

```
Melhor solução -> Geração: 11 Valor: 309 Espaço: 165
Cromossomo: ['1', '1', '1', '1', '0', '1', '0', '0', '0', '0']
Nome: Item 1 R$ 92
Nome: Item 2 R$ 57
Nome: Item 3 R$ 49
Nome: Item 4 R$ 68
Nome: Item 6 R$ 43
Duração da operação: 0.056658
```

Fonte: Acervo Próprio (2019).

Figura 13 – Gráfico da solução para o caso P01 encontrada pelo AG



Fonte: Acervo Próprio (2019).

O resultado exibido na Figura 12 demonstra que a melhor solução para o problema foi obtida na décima primeira geração, possuindo valor de utilidade e de espaço iguais a 309 e 165

respectivamente. Além disso é exibida a representação do cromossomo da melhor resposta, que como definido anteriormente, carrega consigo a informação de quais itens estão sendo levados expressos na forma de uma sequência binária. Logo abaixo os itens que foram selecionados para serem levados no caminhão são exibidos, contendo em sua descrição o nome do item e seu valor de utilidade. Como ultimo item presente na exibição da melhor solução é mostrado o tempo de processamento do algoritmo em uma escala de segundos.

O gráfico representado na Figura 13 mostra o desenvolvimento das soluções, utilizando os valores de utilidade da melhor solução encontrada em cada geração ao longo do avanço das gerações. Através dele, é possível observar o comportamento do algoritmo ao longo de seu processamento em sua busca pela melhor solução possível do problema. O gráfico é composto no eixo das abscissas pelo número correspondente a geração em que se encontra o processamento do algoritmo, e no eixo das ordenadas pela informação do valor de utilidade da melhor resposta encontrada em função do número da geração. Os pontos que compõem o gráfico exibem, portanto, para um dado número de geração em que o algoritmo se encontra, qual o valor de utilidade da melhor solução encontrada naquele dado momento.

Utilizando essas informações pode se observar que, ao longo do passar das gerações no gráfico da Figura 13, a melhor solução está representada por um pico ainda nas primeiras gerações, afinal a melhor solução para o caso foi encontrada na décima primeira geração. Contudo, é notável que o gráfico a partir deste momento contém algumas perturbações até entrar convergir para um resultado constante pouco após a quadragésima geração. Nesse intervalo as perturbações são representações das tentativas do algoritmo de buscar por novas soluções, utilizando as técnicas anteriormente abordadas de cruzamento e mutação, com o intuito de gerar melhores indivíduos. A convergência do gráfico após esse momento mostra que, para o problema montado, o algoritmo não foi capaz de produzir nenhum outro novo indivíduo, que superasse a melhor solução anteriormente encontrada, fazendo com que a população convergisse futuramente para este mesmo valor, embora o resultado obtido seja descrito como a melhor solução para o problema de acordo com o banco de dados utilizado.

Os parâmetros do algoritmo são então alterados na tentativa de obter uma melhor precisão quanto a assertividade do algoritmo, utilizando agora os parâmetros:

Tamanho da População = 20
TESTE 2 Taxa de mutação = 0,02
Número de Gerações = 100

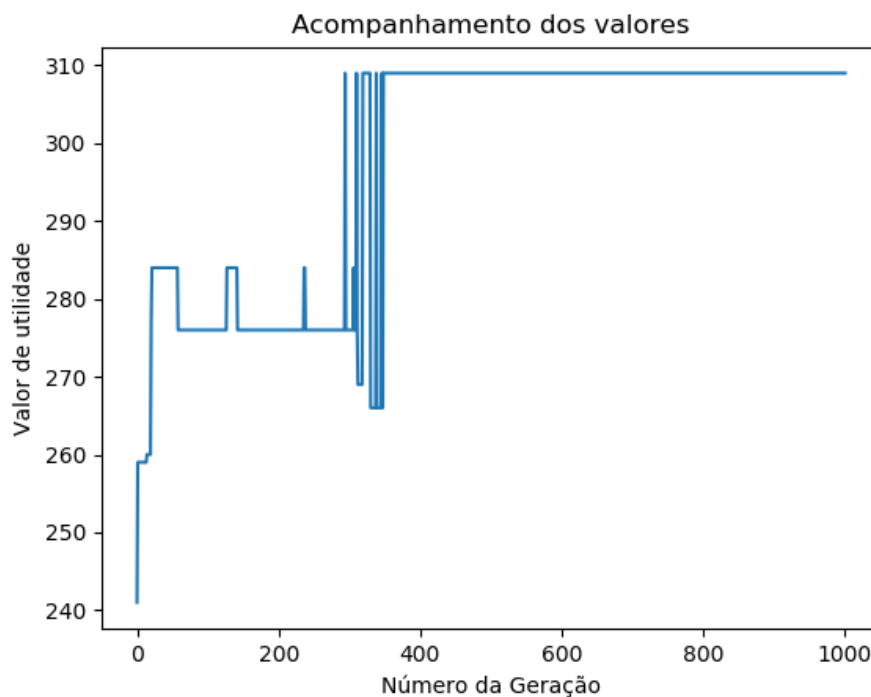
Através dessa manipulação foi possível observar que ao variar o número de gerações a serem processadas pelo AG o resultado ideal ainda é obtido em uma frequência maior do que no experimento anterior, diminuindo assim o número de repetições necessário para que seja encontrada uma resposta ótima para o problema. Contudo, esta alteração provoca, como era de se esperar, um aumento no tempo de processamento, embora o AG continue trabalhando em um tempo menor que 1 segundo. Os resultados obtidos pelo algoritmo estão representados nas Figuras 14 e 15.

Figura 14 – Solução para o caso P01 encontrada pelo AG

```
Melhor solução -> Geração: 294 Valor: 309 Espaço: 165
Cromossomo: ['1', '1', '1', '1', '0', '1', '0', '0', '0', '0']
Nome: Item 1 R$ 92
Nome: Item 2 R$ 57
Nome: Item 3 R$ 49
Nome: Item 4 R$ 68
Nome: Item 6 R$ 43
Duração da operação: 0.524429
```

Fonte: Acervo Próprio (2019).

Figura 15 – Gráfico da solução para o caso P01 encontrada pelo AG



Fonte: Acervo Próprio (2019).

Assim como no caso anterior, a melhor solução encontrada pelo algoritmo é primeiramente detalhada na como mostra a Figura 14, e logo após é exibido o gráfico de acompanhamento dos valores para o processamento em questão. Através da curva exibida na Figura 15 é possível notar que o algoritmo, nas primeiras 50 gerações, encontrou uma solução de valor de utilidade igual a 285 aproximadamente, contudo, através das técnicas de cruzamento e mutação já comentadas, conseguiu logo após a geração 300 aproximadamente encontrar um indivíduo que representa uma melhor solução ao problema, e então a mesma convergência de valores se repete no gráfico a partir da geração 380 aproximadamente, como visto no caso anterior.

É notável o comportamento heurístico do algoritmo ao se observar o teste feito com um valor de população igual a 100 e o outro igual a 1000, onde, para os dois casos, a melhor solução foi encontrada em uma geração de ordem relativamente baixa em função do todo. Para o primeiro caso o melhor indivíduo foi encontrado na geração 11 de 100 e para o segundo experimento foi encontrado na geração 294 de 1000.

5.4. TESTE 2 – Carregamento de Caminhão

Uma vez tendo aferido a eficácia do algoritmo construído com um exemplo fornecido pelo *Data for the 01 Knapsack Problem*, é possível aplicar o AG sobre uma situação montada, que faz alusão a um caso real, para analisar o comportamento e resultados obtidos pela ferramenta sobre determinadas condições. O caso a ser estudado é o citado anteriormente de locação de cargas em um caminhão de volume disponível L .

A situação criada tem como única definição de parâmetros uma lista fictícia de produtos quaisquer, com características de valor de utilidade e espaço ocupado distintas, e o espaço total disponível do caminhão que será definido dentro do algoritmo.

Na Tabela 4 estão listados todos os produtos que estão disponíveis para serem levados no caminhão fictício de espaço/peso limite igual a 100. Alguns dos produtos estão repetidos para simular um caso do problema da mochila inteira, onde a quantidade de cada um dos itens disponíveis também varia, que traz uma representação mais fiel aos casos de carregamento vistos na realidade.

Tabela 4 – Problema da locação de cargas no caminhão

SITUAÇÃO CRIADA – CARREGAMENTO DE CAMINHÃO

| Limite do caminhão | | 100 |
|---------------------------|-----------------------|---------------------------|
| Item | Espaço ocupado | Valor de utilidade |
| TV LED 43" | 9,1 | 1799 |
| Ar Condicionado | 31 | 1038 |
| Cafeteira | 3,6 | 260 |
| Computador | 12 | 1646 |
| Computador | 12 | 1646 |
| Fogão | 32 | 999 |
| Micro-ondas | 16 | 648 |
| Lavadora de Roupas | 34 | 1254 |
| Ventilador | 4,1 | 169 |
| Aspirador de Pó | 5,9 | 389 |
| Liquidificador | 1,8 | 139 |
| Liquidificador | 1,8 | 139 |
| Geladeira | 58 | 2399 |
| Geladeira | 58 | 2399 |
| Vídeo Game | 4,35 | 1535 |
| Vídeo Game | 4,35 | 1535 |
| Vídeo Game | 4,35 | 1535 |
| Notebook | 1,95 | 2008 |
| Smartphone | 0,2 | 2649 |
| Smartphone | 0,2 | 2649 |

Fonte: Acervo Próprio (2019).

Diversos testes foram realizados sobre a situação montada afim de obter o melhor resultado possível para o problema, portanto, foram alterados por diversas vezes os parâmetros de tamanho da população, taxa de mutação e número de gerações.

Os parâmetros usados para execução do algoritmo sobre o problema foram:

| | |
|----------------|----------------------------------|
| | Tamanho da População = 20 |
| TESTE 1 | Taxa de mutação = 0,01 |
| | Número de Gerações = 100 |

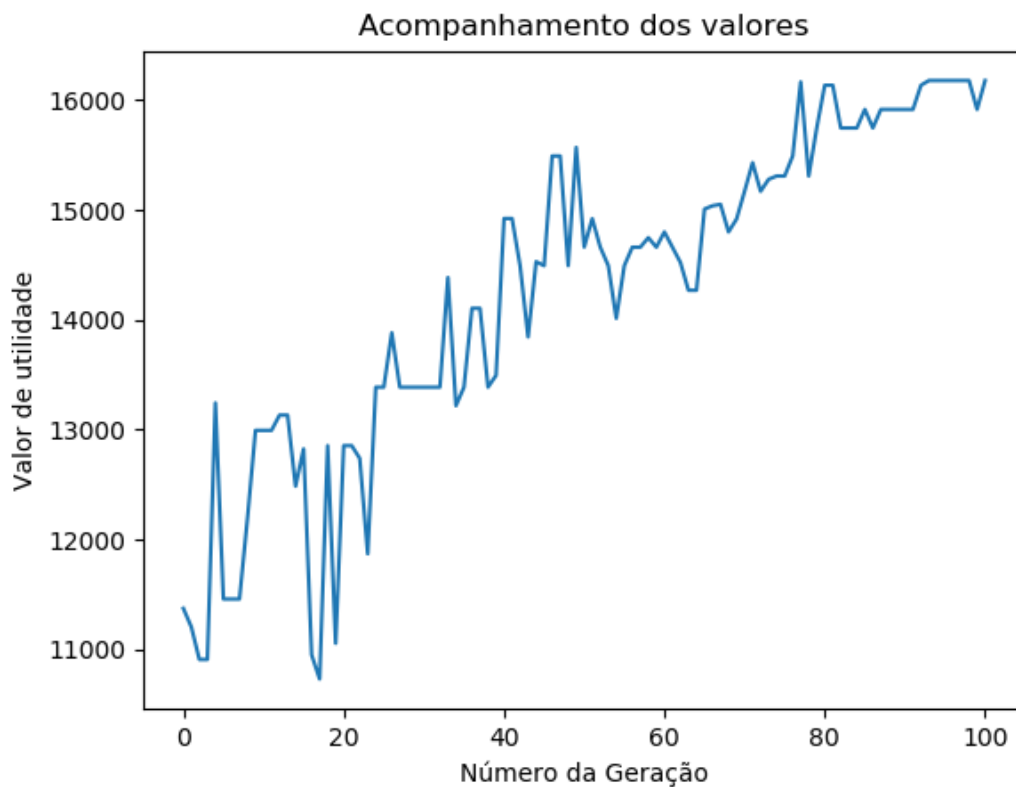
Os resultados obtidos através dos testes assim como os gráficos das respectivas simulações estão exibidos nas Figuras 16 e 17.

Figura 16 – Solução para o caso montado encontrada pelo AG Teste 1

```
Melhor solução -> Geração: 93 Valor: 16178 Espaço: 97.09999999999998
Nome: TV LED 43' R$ 1799
Nome: Ar Condicionado R$ 1038
Nome: Computador R$ 1646
Nome: Computador R$ 1646
Nome: Micro-ondas R$ 648
Nome: Liquidificador R$ 139
Nome: Video Game R$ 1535
Nome: Video Game R$ 1535
Nome: Video Game R$ 1535
Nome: Notebook R$ 2008
Nome: Smartphone R$ 2649
Duração da operação: 0.044264
```

Fonte: Acervo Próprio (2019).

Figura 17 – Gráfico da solução para o caso montado encontrada pelo AG Teste 1



Fonte: Acervo Próprio (2019).

A Figura 16 exibe com detalhes a melhor solução encontrada pelo algoritmo no teste em questão, sobre os dados que a imagem fornece faz necessária uma observação a respeito do tempo de processamento que, apesar de se tratar de um teste com mais dados do que o caso anterior utilizado para aferir o algoritmo, foi obtido nele uma duração de processamento menor que essa última. Essa diferença não esperada, uma vez que com mais dados, em teoria, era esperado que o tempo de processamento fosse maior, contudo, para intervalos tão pequenos de tempo diversos fatores da máquina que está processando influenciam no resultado final, alguns desses fatores são: quantidade de memória já alocada, processos em paralelo com o algoritmo, temperatura da máquina no momento do processamento, entre outros.

Na figura 17 é exibida a curva de acompanhamento dos valores para o Teste 01, através do gráfico é possível observar que ao longo da execução desse teste o algoritmo sofreu com diversas variações de valores, algumas bruscas outras mais suaves, contudo, mesmo com as perturbações os valores de utilidade encontrados foram crescendo continuamente. O comportamento sinuoso do gráfico apenas reforça o caráter cíclico do algoritmo de buscar uma solução melhor a cada novo loop, de modo que, através das técnicas que possui, o programa tenta modificar a sua população de soluções de modo a incrementar a nota de avaliação de seus indivíduos, sendo assim, obter melhores soluções para o problema. O crescimento constante dos valores de utilidade aponta para um direcionamento correto do algoritmo, no sentido das melhores soluções para o problema.

Os parâmetros usados para execução do algoritmo sobre o problema em uma segunda tentativa foram:

Tamanho da População = 20

TESTE 2 Taxa de mutação = 0,01

Número de Gerações = 10000

Nesse teste foi incrementado o valor do número de gerações com o intuito de fazer com que o algoritmo, através do incremento do número de passos ou loops que irá executar, encontre uma melhor solução para o problema, assim como, selecione de modo mais eficiente a melhor solução, uma vez que, o espaço e tentativas de busca será incrementado, possibilitando que o mesmo realize mais combinações.

A Figura 18 mostra os resultados obtidos através dos testes e a Figura 19 exibe o gráfico da respectiva simulação.

Figura 18 – Solução para o caso montado encontrada pelo AG Teste 2

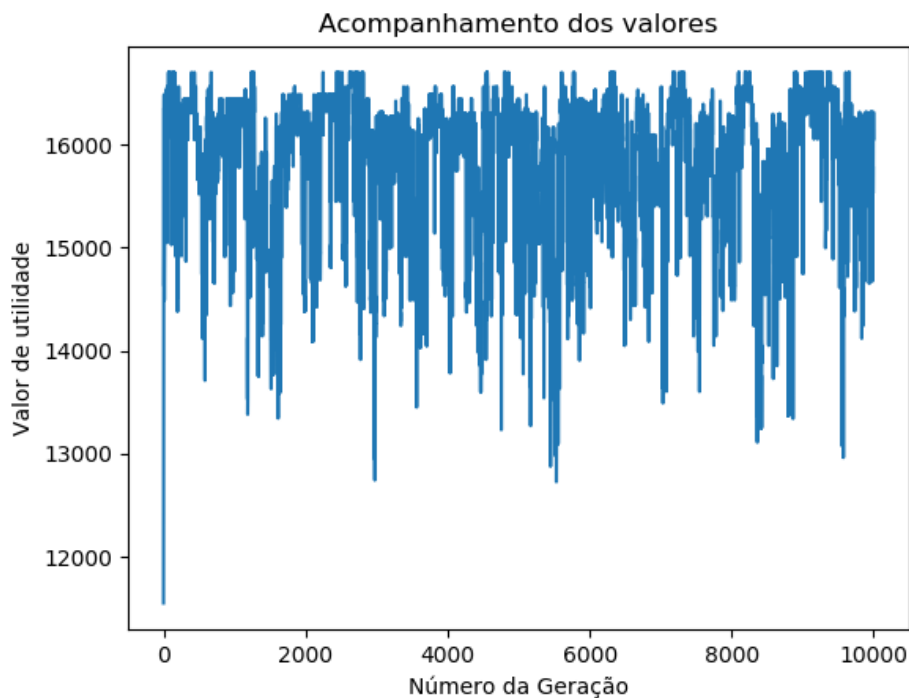
```

Melhor solução -> Geração: 68 Valor: 16703 Espaço: 99.49999999999999
Nome: TV LED 43' R$ 1799
Nome: Cafeteira R$ 260
Nome: Computador R$ 1646
Nome: Computador R$ 1646
Nome: Lavadora de Roupas R$ 1254
Nome: Ventilador R$ 169
Nome: Aspirador de Pó R$ 389
Nome: Liquidificador R$ 139
Nome: Liquidificador R$ 139
Nome: Video Game R$ 1535
Nome: Video Game R$ 1535
Nome: Video Game R$ 1535
Nome: Notebook R$ 2008
Nome: Smartphone R$ 2649
Duração da operação: 5.194111

```

Fonte: Acervo Próprio (2019).

Figura 19 – Gráfico da solução para o caso montado encontrada pelo AG Teste 2



Fonte: Acervo Próprio (2019).

O resultado obtido para a melhor solução encontrada no teste é novamente detalhado, sendo exibido na Figura 18, novamente é notável o tempo de processamento obtido, tendo alcançado agora o valor de 5,194 segundos aproximadamente. De modo inesperado a melhor

solução encontrada para esse teste foi obtida ainda na geração 68, um valor muito pequeno se comparadas as 10000 gerações executadas pelo algoritmo.

Em se tratando do gráfico de acompanhamento de valores produzido pelo teste, é notável a quantidade de perturbações presentes no mesmo, o que aumenta drasticamente sua sinuosidade. Esse comportamento novamente reforça seu caráter de busca, sempre realizando modificações em seus dados na tentativa de obter uma solução melhor. Contudo, um elevado número de gerações torna difícil a visualização e acompanhamento do desempenho do algoritmo ao longo do processo.

Os resultados mostrados nas Figuras 18 e 19 são bem explícitos com relação a proximidade das soluções, o que indica que o algoritmo possui uma certa tendência a obter sempre o resultado mais otimizado possível, contudo, existem outros detalhes presentes nas simulações que são notáveis para os objetivos desse trabalho.

Ao analisar algumas repetições da execução do algoritmo nas configurações expostas anteriormente, onde havia a diferença no tamanho da população de 100 para 10000 para os casos Teste 1 e Teste 2 respectivamente, pode-se notar que ao incrementar esse valor o algoritmo, estando munido de um espaço de repetições e tentativas maior de busca, se sobressai em encontrar a melhor resposta para o problema.

Para o caso em questão a melhor resposta após repetidos testes foi a obtida na Figura 18, onde o valor de utilidade total do indivíduo é de 16703 e o espaço ocupado/peso é de 99,4999. Essa solução pode ser obtida em testes com tamanho de população menor, contudo, a mesma aparece após algumas poucas execuções do algoritmo, porém o mesmo não ocorre ao aumentar o valor do tamanho da população. Apesar da grandeza deste número ter feito com que o algoritmo finalizasse o processo em um tempo consideravelmente maior em comparação com o experimento de tamanho de população igual a 100, o tempo em que foi executado (Duração 5,1941 segundos) é consideravelmente rápido, uma vez que o processo acaba por retornar o que seria a melhor solução possível utilizando essa técnica, em uma faixa de assertividade muito maior como comprovado empiricamente.

5.5. COMENTÁRIOS

Neste capítulo foram apresentados os testes e resultados obtidos por meio da aplicação da metodologia proposta e desenvolvida no Capítulo 2 (O Problema da Mochila) em conjunto com a estratégia tecnológica do Algoritmo Genético (Capítulo 3). O objetivo destes testes foi avaliar qual a melhor solução do PM, com as devidas adaptações de modo a se tornar semelhante a um problema envolvendo o carregamento de um caminhão.

6. CONCLUSÃO

Analisando os resultados, que foram exibidos de maneira que o usuário possa avaliar o comportamento do algoritmo como um todo, ao longo do problema, algumas conclusões podem ser tomadas a respeito dos AGs como ferramenta de solução para o problema, suas particularidades e características influenciadas pela linguagem de programação utilizada.

Por meio da análise do comportamento da ferramenta, com o decorrer do processamento é notável como o sistema reproduz de maneira semelhante o processo de evolução natural. Os resultados mostrados em alguns dos gráficos exibem, por meio de suas curvas, variações nas soluções encontradas pelo algoritmo, em especial na Figura 17, a variação mostrada tem caráter crescente e denota uma busca por evolução contínua do algoritmo até o término do processamento.

Através dos gráficos mostrados nas Figuras 13 e 15 é possível notar, devido a suavidade dos dados, que quando os AGs se depararam com os indivíduos correspondentes a solução ideal para o problema, houveram algumas tentativas malsucedidas de busca por soluções ainda melhores, conforme representado pelas variações mostradas pelos picos alternados. Essa característica é benéfica para o algoritmo pois denota sua capacidade estatística de contínua busca por melhores soluções através do uso dos operadores de cruzamento e de mutação.

O tempo de execução do processamento do algoritmo, mostrado ao final de cada execução, representa para a ferramenta uma das variáveis mais importantes de seu funcionamento, uma vez que, em uma aplicação real o operador teria de lidar com cargas imensamente maiores. Esse aumento do espaço amostral causaria grande impacto no tempo de processamento do algoritmo, tornando a solução do problema mais lenta. A utilização de certos elementos dentro deste trabalho tem o intuito de contribuir para a diminuição desse tempo, uma vez que trazem agilidade ao AG. Uma destas técnicas é a modelagem dos itens do problema de maneira direta, onde suas características se resumem a apenas dois atributos (peso e valor de utilidade), e são lançadas ao algoritmo apenas na sua etapa de avaliação, retirando assim o fardo de que estes dois atributos estejam sempre presentes no processamento ocupando espaços desnecessários na memória, fazendo com que o algoritmo seja capaz de trabalhar mais agilmente com estes dados.

É notável também que a escolha da linguagem de programação influenciou grandemente no desenvolvimento e eficácia da ferramenta. O Python traz inúmeras vantagens para programadores que realizam pesquisas e aplicações utilizando Inteligência Artificial e suas ramificações, e seu poder de rapidez de processamento de dados recebeu um considerável

incremento nos últimos anos, e atualmente, apesar de ser uma linguagem de alto nível, não deixa em nada a desejar em relação a linguagens de mais baixo nível, como o C, chegando até mesmo a superá-las em diversas comparações.

Em resumo o AG se mostrou uma poderosa ferramenta na resolução de problemas combinatórios impossíveis por métodos comuns, o que eleva ainda mais seu valor como metodologia, pois grandes partes dos problemas logísticos, de transportes, de processos industriais têm impermeados em sua natureza problema combinatórios. A solução rápida do problema aqui descrito, permite ainda classificar como eficaz a ferramenta modelada não somente para o caso montado para estudo, mas como também para exemplos semelhantes de aplicação.

6.1. TRABALHOS FUTUROS

Como sugestão de trabalhos futuros utilizando a ferramenta e sua aplicação demonstradas neste trabalho pode-se citar a aplicação do algoritmo na solução de um problema completo de logística, onde todo o processo de transporte de carga, fluxo de mercadorias, controle de entregas, dentre outros processos semelhantes ficaria a cargo do algoritmo decidir qual o caminho mais eficiente a ser tomado dentre as diversas possibilidades presentes nestes processos. A ferramenta aqui mostrada, pode ainda, ser incrementada com o advento de diversos elementos técnicos, uma vez que, se fornecidos e modelados corretamente os dados, o AG encontrará uma solução ótima para qualquer problema combinatório.

Outra sugestão seria o desenvolvimento do próprio algoritmo de modo a torna-lo uma ferramenta mais completa, intuitiva e ilustrativa, adicionando ao seu funcionamento uma interface clara e direta para que qualquer usuário possa operar assim como manipular a exibição dos resultados de modo a devolver dados mais ilustrados e personalizados para o fácil entendimento do operador.

A ferramenta criada também pode ser aplicada nos ramos da Engenharia Elétrica, mais em específico aos problemas relativos ao Sistemas Elétricos de Potência (SEP), utilizando a relação entre o PM e os problemas de fluxo de carga na operação destes sistemas para o desenvolvimento de uma ferramenta de reposta rápida a possíveis surtos na rede, ou que auxiliasse o trabalho de gerencia dos operadores.

Como constado, a aplicação dos AGs não se limita a Engenharia Elétrica. Dessa forma, é imprescindível que se amplie seu estudo em todos os sentidos, permitindo grandes contribuições à sociedade, papel fundamental da comunidade científica.

Referência Bibliográfica

A COMPARATIVE Study of Adaptive Crossover Operators for Genetic Algorithms to Resolve the Traveling Salesman Problem. Kenitra, Morocco: [s.n.], 2011. Disponível em: <<https://arxiv.org/ftp/arxiv/papers/1203/1203.3097.pdf>>. Acesso em: 12 set. 2018

ALMEIDA, Charles Ornelas et al. **Problemas N P-Completo e Algoritmos Aproximados.** 28 set. 2010. 78 slides. Material apresentado para a disciplina de Projeto de Algoritmos.

BEASLEY, J.E. A genetic algorithm for the set covering problem. **European Journal of Operational Research**, London, UK, 25 out. 1996. Volume 94, p. 392-404. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/037722179500159X#!>>. Acesso em: 04 set. 2018.

BURIOL, Luciana Salete. **Algoritmo memético para o problema do caixeiro viajante assimétrico como parte de um framework para algoritmos evolutivos.** 2000. 136 p. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Estadual de Campinas, Universidade Estadual de Campinas, Campinas, SP, 2000.

COMPUTAÇÃO Evolutiva: Uma Abordagem Pragmática. Campinas, SP: [s.n.], 2007. Disponível em: <http://ftp://calhau.dca.fee.unicamp.br/pub/docs/vonzuben/ia013_1s07/tutorial EC.pdf>. Acesso em: 20 set. 2018.

CONGRESSO RIBIE, 4., 1998, Brasília. **JOGOS EDUCATIVOS COMPUTADORIZADOS UTILIZANDO A ABORDAGEM DE ALGORITMOS GENÉTICOS...** Brasília, BR: [s.n.], 1998. 13 p. Disponível em: <http://www.ufrgs.br/niece/eventos/RIBIE/1998/pdf/com_pos_dem/151.pdf>. Acesso em: 15 ago. 2018.

DOWNEY, Allen B. **Pense em Python.** 1. ed. [S.l.]: Novatec, 2016. 312 p.

Fogel D.B., Fogel L.J. (1996) **An introduction to evolutionary programming.** In: Alliot JM., Lutton E., Ronald E., Schoenauer M., Snyers D. (eds) Artificial Evolution. AE 1995. Lecture Notes in Computer Science, vol 1063. Springer, Berlin, Heidelberg

GOLDBERG, David E. **Genetic Algorithms in Search, Optimization and Machine Learning.** 1. ed. Boston, USA: MIT Press, 1989. 372 p.

LINDEN, Ricardo. **Algoritmos Genéticos.** 3. ed. [S.l.]: Ciência Moderna Ltda, 2012. 496 p.

MENEZES, Nilo Ney Coutinho. **Introdução à Programação com Python.** 2. ed. [S.l.]: Novatec, 2014. 334 p.

MICHALEWICZ, Zbigniew. **Genetic Algorithms + Data Structures = Evolution Programs.** 3. ed. [S.l.]: Springer, 1996. 387 p.

MITCHELL, Melanie. **An Introduction to Genetic Algorithms.** 5. ed. London, England: MIT Press, 1998. 162 p.

POLI, Riccardo; LANGDON, William B.; MCPHEE, Nicholas Freitag. **A Field Guide to Genetic Programming**. 1. ed. [S.l.]: Lulu Enterprises, 2008. 252 p.

SHEPPARD, Clinton. **Genetic Algorithms with Python**. 1. ed. [S.l.]: CreateSpace Independent Publishing Platform, 2016. 532 p.

APÊNDICE A – PUBLICAÇÕES

Além do desenvolvimento de toda implementação descrita neste texto de Trabalho de Conclusão de Curso, foi desenvolvido em paralelo, porém dentro da linha de pesquisa do grupo, dois artigos.

1. SILVA, J. C.; MOREIRA, L. F. R. and FREITAS, S. C. L. “Analysis of the Energy Quality of the Electronic Laboratory of the Federal University of Tocantins Using Artificial Immunological Systems”, SpringerNature’s proceedings, v. 01, p. 01, 2018.

2. ALVARENGA, R. F.; SALES, A. G. B.; GOMES, V. C. O.; MOREIRA, L. F. R.; SILVA, J. C.; FREITAS, S. C. L.; OLIVEIRA, P. S. “Classification of Disturb Voltage in Electricity Distribution Systems Using Artificial Immunological Systems”, Sodebras proceeding, v. 165, 2019.