

UNIVERSIDADE FEDERAL DE ALAGOAS - UFAL  
INSTITUTO DE COMPUTAÇÃO - IC  
CIÊNCIA DA COMPUTAÇÃO

JOSÉ RUBENS DA SILVA BRITO

**UM OPERADOR DE CRUZAMENTO BASEADO NA PRESERVAÇÃO DOS  
BLOCOS DE CONSTRUÇÃO**

Maceió - AL  
2022

JOSÉ RUBENS DA SILVA BRITO

**UM OPERADOR DE CRUZAMENTO BASEADO NA PRESERVAÇÃO DOS  
BLOCOS DE CONSTRUÇÃO**

Trabalho de Conclusão de Curso  
submetido ao corpo docente do Instituto  
de Computação da Universidade Federal  
de Alagoas. Orientado pela Profa. Dra.  
Roberta Vilhena Vieira Lopes.

Maceió - AL  
2022

**Catlogação na fonte**  
**Universidade Federal de Alagoas**  
**Biblioteca Central**  
**Divisão de Tratamento Técnico**

Bibliotecária Responsável: Maria Helena Mendes Lessa– CRB4 - 1616

B862u Brito, José Rubens da Silva.  
Um operador de cruzamento baseado na preservação dos blocos de construção / José Rubens da Silva Brito. – 2023.  
63 f. : il. color.

Orientadora: Roberta Vilhena Vieira Lopes.  
Monografia (Trabalho de Conclusão de Curso em Ciência da Computação) – Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2022.

Bibliografia: f. 52-56.  
Apêndices: f. 57-63.

1. Algoritmos genéticos. 2. Algoritmo de Holland. 3. Operador de cruzamento. 4. Teoria dos *schemata*. I. Título.

CDU: 004.421

## **Folha de Aprovação**

AUTOR: JOSÉ RUBENS DA SILVA BRITO

### **UM OPERADOR DE CRUZAMENTO BASEADO NA PRESERVAÇÃO DOS BLOCOS DE CONSTRUÇÃO**

Trabalho de Conclusão de Curso  
submetido ao corpo docente do Instituto  
de Computação da Universidade Federal  
de Alagoas e aprovado em 14 de julho de  
2022

---

(Profª. Dra. Roberta Vilhena Vieira Lopes - UFAL) (Orientadora)

#### **Banca Examinadora:**

---

(Prof. Dr. Evandro de Barros Costa - UFAL) (Examinador Interno)

---

(Profª. Dra. Andrilene Ferreira Maciel - UFRPE)(Examinador Externo)

## **DEDICATÓRIA**

Dedico esse trabalho aos meus pais e a minha noiva, que sempre me apoiaram e me incentivaram.

## AGRADECIMENTOS

Primeiramente, quero agradecer a Deus, pela minha vida, e por me dar saúde e sabedoria para concluir essa etapa da minha vida.

Aos meus pais Tereza Cristina e Rubens Amaro, que sempre foram meu pilar, e que me incentivaram nos momentos difíceis e compreenderam a minha ausência enquanto eu me dediquei à realização deste trabalho. Ao meu irmão José Rafael, que sempre torceu pelas minhas vitórias.

À minha noiva Alice Maria, que esteve ao meu lado nos momentos mais difíceis desta jornada, demonstrando toda compreensão, paciência e afeto.

À minha orientadora Roberta Lopes, por aceitar me conduzir neste trabalho, bem como por toda paciência, auxílio e compreensão dedicada.

Agradeço ao professor Evandro Costa, pelas oportunidades geradas durante o meu percurso acadêmico.

Agradeço a todo corpo docente do Instituto de Computação, por todos os ensinamentos que me permitiram apresentar um melhor desempenho no meu processo de formação acadêmica e profissional.

Agradeço ao corpo técnico, em especial a Ana Ferreira e Marcelo Gusmão pelo apoio fornecido nas demandas burocráticas.

Agradeço aos meus amigos e colegas de curso, com quem convivi intensamente durante os últimos anos, pelo companheirismo e pela troca de experiências que me permitiram crescer não só como pessoa, mas também como formando. Em especial à Matheus Gomes, Ramon Basto, João Victor e Lucas Lisboa.

Agradeço aos professores que aceitaram compor a banca examinadora.

Agradeço a todos que convivi ao longo desses anos de curso, que me incentivaram e que certamente impactaram na minha formação acadêmica.

Por fim, agradeço a você leitor.

## EPÍGRAFE

*“Machines take me by surprise with great frequency”*

(Alan Turing, ***Computing Machinery And Intelligence***, 1950)

## RESUMO

Neste trabalho, foi introduzido um operador genético de cruzamento baseado na preservação dos blocos de construção que existem entre os cromossomos pais. Esse operador é utilizado no lugar do operador genético de cruzamento de um ponto de corte no algoritmo genético de Holland, onde ocorre a preservação dos blocos de construção que existem nos pais em seus filhos, com o propósito de resolver problemas de otimização. O algoritmo genético com o operador proposto, é comparado como o mesmo algoritmo considerando o cruzamento de um ponto e dois pontos de cortes. Esses três operadores foram testados sobre dois problemas de otimização, caixeiro viajante e *OneMax*. Para avaliar o desempenho do operador de cruzamento proposto em relação ao operador de cruzamento genético de um ponto de corte, onde serão realizados dois tipos de análise: convergência do algoritmo genético e qualidade dos cromossomos encontrados. Ao fim dos testes, foram analisados os resultados obtidos pelo operador proposto com um ponto de corte, e com dois pontos, e o operador genético de cruzamento de um ponto de corte do algoritmo genético de Holland.

**Palavras-chaves:** Algoritmos genéticos, Algoritmo de Holland, operador de cruzamento, teoria dos *schemata*

## ABSTRACT

In this work, a genetic crossover operator based on the preservation of the building blocks that exist between the parent chromosomes has been introduced. This operator is used in place of the genetic crossover operator of a cutoff point in Holland's genetic algorithm, where the preservation of the building blocks that exist in the parents in their children occurs, with the purpose of solving optimization problems. The genetic algorithm with the proposed operator, is compared with the same algorithm considering one cross cut point and two cut points. These three operators were tested on two optimization problems, traveling salesman and OneMax. To evaluate the performance of the proposed crossover operator compared to the genetic one-cutting point cross operator, where two types of analysis will be performed: convergence of the genetic algorithm and quality of the chromosomes found. At the end of the tests, the results obtained by the proposed operator with one cutoff point, and with two points, and the genetic crossover operator of one cutoff point of Holland's genetic algorithm were analyzed.

**Keywords:** Genetic Algorithms, Holland's algorithm, crossover operator, schemata theory

## LISTA DE FIGURAS

Figura 2.1 - Máximo local e Máximo global em uma função hipotética .....	18
Figura 2.2 - Vetor binário com duas posições.....	19
Figura 2.3 - Vetor binário com três posições.....	20
Figura 2.4 - Operador Genético de Cruzamento.....	22
Figura 2.5 - Operador Genético de Mutação.....	23
Figura 2.6 - Operador Genético de Seleção.....	24

## LISTA DE GRÁFICOS

Gráfico 4.1 - Resultado AG de Holland para OneMax.....	42
Gráfico 4.2 - Resultado cruzamento proposto com um ponto de corte para OneMax.....	42
Gráfico 4.3 - Resultado cruzamento proposto com dois pontos de cortes para OneMax.....	43
Gráfico 4.4 - Resultado AG de Holland para o Caixeiro Viajante.....	46
Gráfico 4.5 - Resultado cruzamento proposto com um ponto de corte para o Caixeiro Viajante.....	46
Gráfico 4.6- Resultado cruzamento proposto com dois pontos de cortes para o Caixeiro Viajante.....	47

## LISTA DE QUADRO

Quadro 4.1 - Combinação dos parâmetros para o algoritmo genético de Holland.....	37
Quadro 4.2 - Soluções encontradas para o problema <i>OneMax</i> .....	44
Quadro 4.3 - Soluções encontradas para o problema Caixeiro Viajante.....	48

## LISTA DE ABREVIATURAS

AG - Algoritmos Genéticos

AE - Algoritmos Evolucionários

RNA - Redes Neurais Artificiais

$\vec{i}_m^1$  - Vetor de dimensão  $m$

$\hat{o}$  - Função objetiva de um problema

$D$  - Domínio de um problema

*decod* - Função que recebe um cromossomo e retorna o elemento do domínio do problema que este cromossomo representa

$f$  - Função de adaptação do algoritmo genético de Holland e suas variações

$C_u$  - Conjunto dos cromossomos cuja representação adotada é  $u$  do algoritmo genético de Holland e suas variações

$p$  - População do algoritmo genético de Holland e suas variações

$m$  - Tamanho da população no algoritmo genético de Holland e suas variações

$\check{p}$  - População dos cromossomos selecionados para gerar descendentes no algoritmo genético de Holland e suas variações

$prob_c$  - Probabilidade de cruzamento no algoritmo genético de Holland e suas variações

$prob_m$  - Probabilidade de cruzamento no algoritmo genético de Holland e suas variações

*one* - Função de cruzamento de um ponto de corte

*flip* - Função de mutação por complemento

*elite* - Função de seleção elitista

$s$  - É um *schemata*

$s_j$  - Elemento do *schemata*  $s$

$v_j$  - Variável do *schemata*  $s$

$\vec{s}$  - Espaço de busca representado pelo *schemata*  $s$

$const(s)$  - Função posição das constantes no *schemata*  $s$

$var(s)$  - Função posição das variáveis no *schemata*  $s$

$\Phi(s, P)$  - Função de adaptação do *schemata*  $s$  na população  $P$

$w$  - Número de variáveis de um *schemata*

$\xi(s, P)$  - Função número de indivíduos que pertencem ao *schemata*  $s$  na população  $p$

$card(X)$  - Função cardinalidade do conjunto  $X$

$p_{\Delta}$  - Probabilidade de um *schemata* ser preservado pela ação dos operadores genéticos de cruzamento

$p_{\Omega}$  - Probabilidade de um *schemata* ser preservado pela ação dos operadores genéticos de mutação

$p_{\nabla}$  - Probabilidade de um *schemata* ser preservado pela ação dos operadores genéticos de inversão

$\Delta$  - Distância entre a primeira e a última ocorrência de uma constante em um *schemata*

## SUMÁRIO

<b>INTRODUÇÃO</b>	<b>13</b>
1.1 - TEMA	14
1.2 - CONTEXTUALIZAÇÃO E PROBLEMA	14
1.3 - OBJETIVOS	15
1.3.1 - GERAIS	15
1.3.2 - ESPECÍFICOS	15
1.4 - JUSTIFICATIVA	15
1.5 - ESTRUTURA DO TRABALHO	16
<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>17</b>
2.1 - ALGORITMO GENÉTICO DE HOLLAND	17
2.1.1 - REPRESENTAÇÃO DO INDIVÍDUO	19
2.1.2 - FUNÇÃO DE ADAPTAÇÃO	20
2.1.3 - POPULAÇÃO	20
2.1.4 - OPERADORES GENÉTICOS	21
2.1.4.1 - OPERADOR GENÉTICO DE CRUZAMENTO	21
2.1.4.1.1 - CRUZAMENTO DE UM PONTO DE CORTE	21
2.1.4.2 - OPERADOR GENÉTICO DE MUTAÇÃO	22
2.1.4.2.1 - MUTAÇÃO POR COMPLEMENTO	22
2.1.4.3 - SELEÇÃO	23
2.1.4.3.1 - SELEÇÃO ELITISTA	23
2.1.5 - PARÂMETROS DOS ALGORITMOS GENÉTICOS	23
2.1.5.1 - TAXA DE CRUZAMENTO	24
2.1.5.2 - TAXA DE MUTAÇÃO	24
2.2 - TRABALHO CORRELATOS	25
<b>UM NOVO MODELO DE OPERADOR DE CRUZAMENTO BASEADO NA PRESERVAÇÃO DE BLOCOS DE CONSTRUÇÃO</b>	<b>27</b>
3.1 - INTRODUÇÃO	27

3.3 - CRUZAMENTO PROPOSTO	33
<b>MATERIAIS E RESULTADOS</b>	<b>35</b>
4.1 - MATERIAIS E MÉTODOS	35
4.1.1 - AMBIENTE DE TESTE	35
4.1.2 - ALGORITMO GENÉTICO DESENVOLVIDO	35
4.1.2.1 - ESPECIFICAÇÕES GERAIS	35
4.1.2.2 - ESPECIFICAÇÃO DOS ALGORITMOS	36
4.1.3 - MÉTRICAS DE AVALIAÇÃO	36
4.1.4 - PROBLEMAS PROPOSTO	37
4.1.4.1 - ONEMAX	37
4.1.4.2 - CAIXEIRO VIAJANTE	38
4.2 - RESULTADOS E DISCUSSÃO	40
4.2.1 PROBLEMA ONEMAX	40
4.2.2 - PROBLEMA DO CAIXEIRO VIAJANTE	44
<b>CONCLUSÃO</b>	<b>49</b>
5.1 TRABALHOS FUTUROS	50
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>51</b>

# CAPÍTULO 1

## INTRODUÇÃO

No período entre 1950 e 1960, muitos cientistas da computação iniciaram os estudos sobre os sistemas evolutivos, como resultado eles desenvolveram várias abordagens computacionais que simulavam o processo de evolução das espécies, para serem usadas como ferramenta de otimização (MITCHELL, 1998), denominadas de algoritmos evolucionários. Tendo como ideia central, a metáfora de que o processo de evolução das espécies é um processo de otimização dos indivíduos pertencentes a essa espécie, sendo os indivíduos sobreviventes os mais adaptados para o seu habitat.

Os algoritmos evolucionários têm em comum o fato de trabalharem a evolução de uma população de indivíduos, considerados resultados para um determinado problema, os quais transmitem aos seus descendentes parte de seu material genético. O processo de transmissão do material genético ocorre devido a ação dos operadores genéticos. E o processo de evolução funciona como um filtro que a todo instante determina quem irá nascer, permanecer ou morrer na população.

Os algoritmos evolucionários podem se diferenciar em função da representação do indivíduo, do comportamento dos operadores genéticos e dos filtros usados para gerenciamento da população. Entre esses algoritmos pode-se citar:

- Estratégia evolucionária, proposta por Rechenberg para ser aplicada a problemas contínuos de otimização paramétrica (MITCHELL, 1998).
- Algoritmo genético, desenvolvido por John Holland para achar o resultado de sistemas adaptativos complexos, onde o resultado encontrado depende da interação independente de agentes racionais (MITCHELL, 1998).
- Programação evolutiva, concebida por Fogel para simular a evolução do processo de aprendizagem de uma pessoa (MITCHELL, 1998).

- Programação genética, modelada por Koza para sugerir um conjunto de programas de computador capazes de resolver um determinado problema (KOZA, 1994).

## 1.1 - TEMA

Sendo o algoritmo genético (AG) a abordagem que popularizou os algoritmos evolucionários no meio acadêmico, devido a sua simplicidade e a explicação de natureza probabilística (teoria do *schema*) gerada por Holland para formalizar o comportamento de cada iteração do seu algoritmo.

Os AG se baseiam na teoria da evolução das espécies de Charles Darwin (1959), onde pela ação da pressão sexual os indivíduos mais adaptados ao ambiente, são preteridos pelas fêmeas da sua espécie, aumentando dessa forma o seu número de descendentes. Assim as características presentes nos indivíduos pais serão transmitidas aos indivíduos filhos, caso ela seja o resultado de um gene puro ou dominante. Uma vez que, apenas os indivíduos mais adaptados conseguem sobreviver ao ambiente.

O AG estruturado por Holland é um método que auxilia no deslocamento de uma população de indivíduos, a uma nova população. Esse deslocamento é realizado através da seleção natural, onde há a ocorrência de operadores genéticos sobre os indivíduos, tais como, cruzamento, mutação e inversão. O indivíduo é representado internamente por um cromossomo, que é a estrutura de dados usada na abstração do resultado do problema, cada posição do cromossomo recebe o nome de gene, e cada gene é uma instância de um alelo particular (MITCHELL, 1998).

## 1.2 - CONTEXTUALIZAÇÃO E PROBLEMA

Propor um operador de cruzamento que garanta o comportamento do processo de transmissão hereditária das características dos pais para os filhos, como definido na teoria da evolução das espécies e provado pelo trabalho de Mendel (1866). Tal operador vai garantir que as características presentes nos pais estejam também presentes no filho. E avaliar o comportamento da convergência e da qualidade do resultado encontrado, através da comparação dos valores obtidos quando se executa um do algoritmo de Holland normal com um algoritmo de Holland alterado pela substituição do operador de cruzamento original pelo operador de de cruzamento proposto.

## **1.3 - OBJETIVOS**

### **1.3.1 - GERAIS**

O objetivo deste trabalho é propor e analisar um novo operador de cruzamento, que realiza a conservação dos blocos de construções presentes nos pais em seus filhos, com o intuito de melhorar a qualidade do resultado encontrado para um problema de otimização.

### **1.3.2 - ESPECÍFICOS**

- Investigar se a qualidade dos indivíduos e a convergência são impactados pela ação dos operadores de cruzamento de um ponto e dois pontos de cortes;
- Desenvolver um operador de cruzamento em que os blocos de construção existentes nos pais passem para os seus filhos;
- Analisar se a conservação dos blocos de construção auxiliam na qualidade dos indivíduos e na convergência do AG;
- Analisar as combinações de parâmetros sobre um problema de otimização funcional e um de otimização combinatorial;
- Comparar os resultados gerados pelo algoritmo genético de holland, com um algoritmo genético onde o operador de cruzamento realiza a conservação dos blocos de construção;

## **1.4 - JUSTIFICATIVA**

Quando o operador de cruzamento é aplicado sobre os pais habilitados para gerar descendentes no algoritmo genético de Holland, pode ocorrer a perda de algumas características comuns presentes nesses pais. Fato este que compromete a otimização realizada pelo algoritmo genético, o que pode ocasionar na diminuição do valor de adaptação dos filhos. Diante disso, esse trabalho propôs um operador de cruzamento que preserva a transmissão hereditária de todas as características comuns existentes entre os pais.

## 1.5 - ESTRUTURA DO TRABALHO

Este trabalho está estruturado da seguinte forma. No capítulo 2, encontra-se a fundamentação teórica sobre a qual o trabalho foi construído, abordando os conceitos de algoritmo genético e suas variações. Ainda neste capítulo, serão abordados em detalhes o comportamento dos operadores genéticos existentes e os parâmetros relevantes para a execução adequada do algoritmo genético.

Já no capítulo 3, encontra-se a fundamentação teórica sobre uma das principais ferramentas deste trabalho. É onde pode-se visualizar a fundamentação formal da teoria do *schema*. Ao adentrar no capítulo, será abordado os blocos de construções, onde tem-se uma definição formal junto a uma explanação da sua importância para os algoritmos genéticos. E por fim, será explicado o cruzamento proposto neste trabalho, que tem como pilar os blocos de construção.

O capítulo 4 foi dividido em duas partes, na primeira parte encontra-se a metodologia adotada neste trabalho, bem como, ambiente de teste, algoritmo desenvolvido, especificações do algoritmo e outros tópicos relacionados aos métodos utilizados. Já na segunda parte, encontram-se os problemas que serão resolvidos, e os resultados obtidos para um problema de otimização funcional e um de otimização combinatória.

O capítulo 5 é o último deste trabalho, e nele é possível encontrar as conclusões extraídas do trabalho, bem como, recomendações e propostas para trabalhos futuros, a fim de melhorar a técnica proposta neste trabalho.

## CAPÍTULO 2

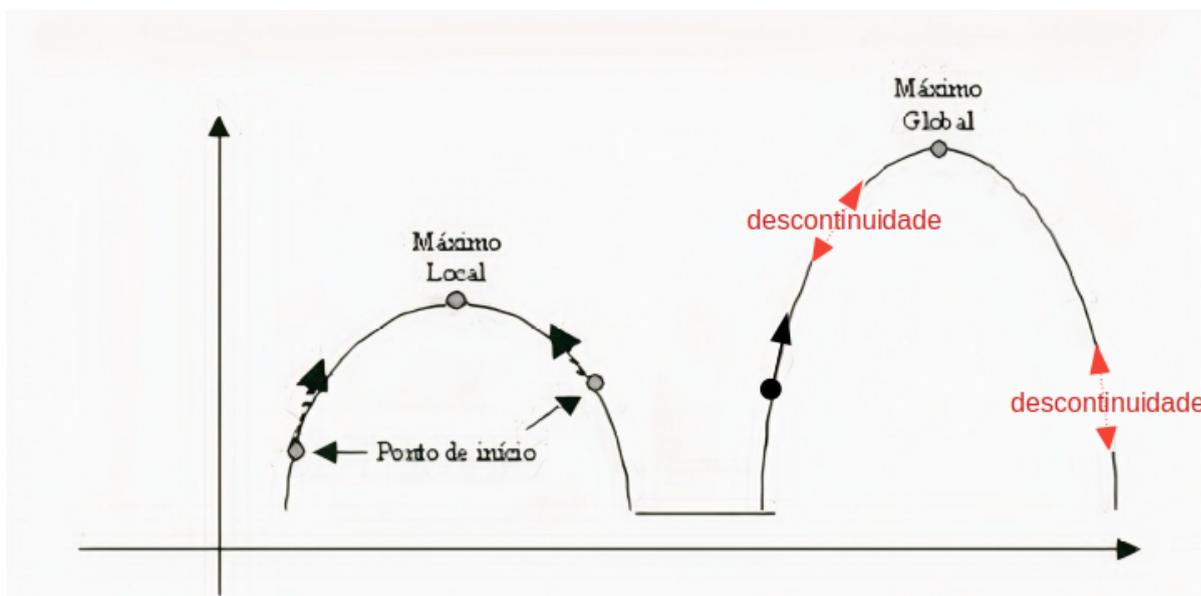
### FUNDAMENTAÇÃO TEÓRICA

Neste capítulo irá ser abordado os conceitos que estão ligados ao assunto deste estudo, seguindo um determinado fluxo, para demonstrar a importância dos blocos de construção para os algoritmos genéticos.

#### 2.1 - ALGORITMO GENÉTICO DE HOLLAND

O algoritmo genético de Holland é um método de otimização de problemas combinatoriais e funcionais. Na otimização de problemas combinatoriais esse algoritmo se comporta como um algoritmo de busca aleatória guiada pela adaptação dos indivíduos pertencentes a população atual (KATO et al, 2021). Enquanto que na otimização de problemas funcionais, esse algoritmo é definido como uma técnica de otimização global, que não faz uso da primeira ou segunda derivada da função como algoritmo *hill climbing*, o que lhe permite fugir dos máximos locais e ser aplicados a funções descontínuas (LINDEN, 2008), pode-se visualizar isso na figura abaixo.

**Figura 2.1** - Máximo local e Máximo global em uma função hipotética



Fonte: Adaptado de Linden (2008)

O algoritmo genético de Holland se trata de um algoritmo iterativo, onde existe a evolução da população de indivíduos, e cada indivíduo representa um possível resultado para o problema. As iterações são denominadas de gerações do AG, são nelas que ocorrem o processo de evolução da população (KATO et al, 2021). No escopo de cada interação, os indivíduos são selecionados para gerar descendentes, submetidos aos operadores genéticos e compõem a nova população. O funcionamento de um AG pode ser visto no pseudocódigo abaixo.

## INÍCIO

**INICIALIZAR** gerar população inicial;

**AVALIAÇÃO** os *fitness* dos indivíduos são calculados ;

**ENQUANTO** ( a condição não for satisfeita )

1 **SELEÇÃO** selecionar os pais;

2 **CRUZAMENTO** selecionar o par de melhores pais;

3 **AVALIAÇÃO** os *fitness* dos novos indivíduos gerados no passo 2 são calculados ;

4 **MUTAÇÃO** muta todos os indivíduos da população;

5 **AVALIAÇÃO** os *fitness* de todos os indivíduos da população;

6 **SELEÇÃO** seleciona os  $x$  indivíduos mais adaptados para a próxima iteração;

**FIM DO ENQUANTO**

**FIM**

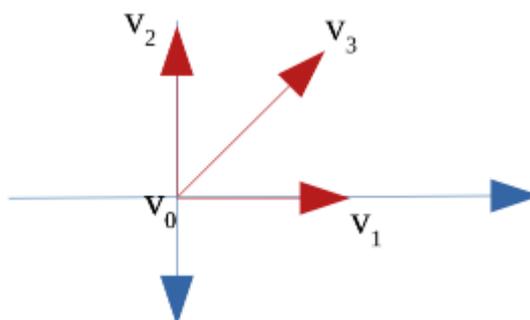
## 2.1.1 - REPRESENTAÇÃO DO INDIVÍDUO

Os indivíduos são abstraídos para dentro do algoritmo por uma estrutura de dados denominada de cromossomo. O cromossomo é um dos componentes fundamentais dos algoritmos genéticos, visto que é nesse momento que começa a modelagem do problema proposto (EIBEN e SMITH, 2015).

O cromossomo de Holland é um vetor binário, fruto da codificação dos elementos pertencentes ao domínio do problema, de modo que esses elementos possam ser manipuláveis pelo computador (EIBEN e SMITH, 2015). Formalmente o cromossomo é definido com um vetor binário de dimensão  $m$ , que armazena o resultado do problema de forma discretizada, sendo definido como  $\vec{i}_m^1 = \langle i_1, i_2, \dots, i_m \rangle$ , onde  $\forall j \in \{1, 2, \dots, m\} (i_j = 0 \vee i_j = 1)$

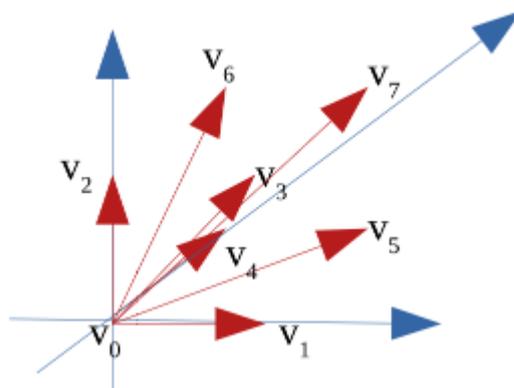
Por exemplo, um vetor com binário duas posições poderia ser representado como na figura 2.2, onde  $v_0 = \langle 00 \rangle$ ,  $v_1 = \langle 10 \rangle$ ,  $v_2 = \langle 01 \rangle$ ,  $v_3 = \langle 11 \rangle$ . Já com três posições poderia ser representado como na figura 2.3, onde  $v_0 = \langle 000 \rangle$ ,  $v_1 = \langle 100 \rangle$ ,  $v_2 = \langle 010 \rangle$ ,  $v_3 = \langle 110 \rangle$ ,  $v_4 = \langle 001 \rangle$ ,  $v_5 = \langle 101 \rangle$ ,  $v_6 = \langle 011 \rangle$  e  $v_7 = \langle 111 \rangle$ ; e assim por diante.

**Figura 2.2** - Vetor binário com duas posições



**Fonte:** (Elaborado pelo Autor, 2022)

**Figura 2.3** - Vetor binário com três posições



**Fonte:** (Elaborado pelo Autor, 2022)

### 2.1.2 - FUNÇÃO DE ADAPTAÇÃO

A função de adaptação deve demonstrar o quanto um indivíduo está adaptado ao problema de tal forma que os indivíduos que pertencem ao domínio do problema recebem um acréscimo positivo na sua adaptação, caso contrário, recebem um acréscimo negativo. Essa função é importante para direcionar a busca do algoritmo genético até a solução ótima global do problema, pois como os algoritmos genéticos são estratégias de maximização, onde, se o cromossomo  $x_1$  representar uma melhor solução que  $x_2$ , então o valor de adaptação do cromossomo  $x_1$  deverá ser maior que o valor da adaptação do cromossomo  $x_2$  (LINDEN, 2008).

Deve-se tomar cuidado quando o problema de otimização é para encontrar o valor minimal. Isso porque os algoritmos genéticos estarão sempre em busca dos cromossomos mais adaptados por definição. Logo para aplicar um algoritmo genético a esses tipo de problema deve ser considerado que o novo valor de adaptação será:

- A negação do valor de adaptação real do problema , ou
- A fração cujo numerador é o número 1 e o denominador é o valor da adaptação real do problema que esse cromossomo representa.

### 2.1.3 - POPULAÇÃO

A população nos AG é um vetor de cromossomos de tamanho finito, que pode ter tamanho fixo ou variável. A população mantém os possíveis resultados para o problema

encontrados até o presente momento pelo algoritmo genético. Servindo de estrutura de entrada para o algoritmo genético (EIBEN e SMITH, 2015).

Dependendo da natureza do problema, a população inicial sobre a qual o algoritmo genético irá trabalhar pode ser gerada aleatoriamente ou lida do ambiente.

A população é definida como um vetor de dimensão  $n$ , onde em cada linha contém um cromossomo, sendo definida como:

$$\vec{p}_1^n = \langle p_1, p_2, \dots, p_n \rangle, \text{ onde, } \forall j \in \{1, 2, \dots, n\} p_j \in \mathcal{C}_v.$$

## 2.1.4 - OPERADORES GENÉTICOS

Os operadores genéticos possuem o princípio de transformar a população  $x$  em um resultado satisfatório, através da produção de novos indivíduos a partir do material genético de seus pais. A ação desses operadores, é denominada por Holland como tensão entre *exploration*<sup>1</sup> e *explotation*<sup>2</sup> (LINDEN, 2008).

Grande parte das ações que os operadores genéticos causam é de teor destrutivo, entretanto, enquadra-se na categoria de exploração, ou seja, ocorre a busca por indivíduos de avaliação melhor que seus pais (LINDEN, 2008).

### 2.1.4.1 - OPERADOR GENÉTICO DE CRUZAMENTO

Esse operador é dito como o principal para gerar diversidade na população (EIBEN e SMITH, 2015), pois é onde ocorre a mistura entre as características dos progenitores. É aplicado de forma probabilística de acordo com a taxa de cruzamento (EIBEN e SMITH, 2015), conseguindo gera novas soluções com adaptação superior, e contribuindo para a convergência genética, ou seja, indo de encontro com algum ponto ótimo, seja local ou global (NUNES, 2018).

O cruzamento é o processo onde um novo indivíduo com potencial para ser a solução do problema é criado, com as informações contidas nos progenitores (EIBEN e SMITH, 2015). Esse processo é realizado através do cruzamento entre dois ou mais progenitores selecionados de forma aleatória ou por métodos de seleção, onde é definido um ponto de corte (ou mais) randômico, em seguida, será gerado os descendentes.

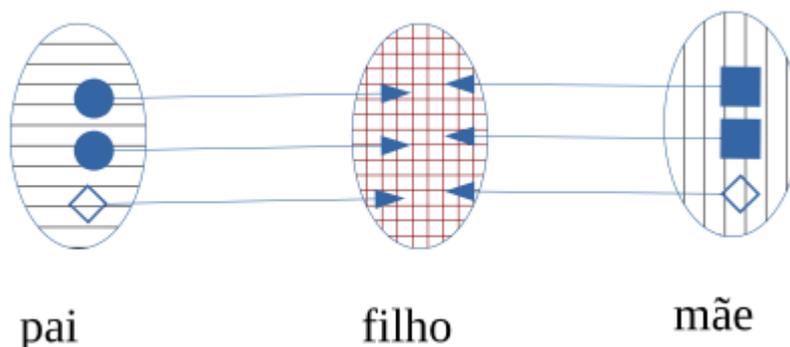
O operador genético de cruzamento pode ser representado pela figura 2.4, onde o material genético de um pai e de uma mãe, são extraídos para formar um filho.

---

<sup>1</sup> A busca por novas adaptações

<sup>2</sup> A manutenção das adaptações úteis feitas até a geração atual

**Figura 2.4 - Operador Genético de Cruzamento**



**Fonte:** (Elaborado pelo Autor, 2022)

### 2.1.4.1.1 - CRUZAMENTO DE UM PONTO DE CORTE

O operador de um ponto de corte recebe um par de cromossomos da população  $\check{p}$ , um ponto de corte  $j \in \{1, \dots, m\}$  e um número randômico  $r_c \in [0, 100]$ , e retornar o par de cromossomos fornecidos, se  $r_c$  é maior que a probabilidade de cruzamento  $prob_c$ ; caso contrário, retornar um novo par de cromossomos que são formados pela combinação dos vetores dos cromossomos fornecidos.

Esse cruzamento é uma função *one* definido como:

$$one(\vec{x}_1^m, \vec{y}_1^m, j, r_c) = (\vec{z}_1^m, \vec{w}_1^m), \text{ onde } \vec{z}_1^j = \vec{x}_1^j, \vec{z}_{j+1}^m = \vec{y}_{j+1}^m, \vec{w}_1^j = \vec{y}_1^j \text{ e } \vec{w}_{j+1}^m = \vec{x}_{j+1}^m$$

Supondo que a função recebe os cromossomos  $x_1$  e  $x_2$ , é gerado aleatoriamente um ponto de corte  $c \in [1, len(x_1)]$ . Caso o  $r_c$  seja menor que a  $prob_c$  definida, então é gerado dois filhos através do cruzamento entre o  $x_1$  e  $x_2$ .

O cruzamento de um ponto de corte, possui uma desvantagem, como o ponto de corte é definido de forma aleatória, ele pode quebrar os *schemata* que o indivíduo possui. Quanto maior for o comprimento do *schemata* maior a probabilidade dele ser destruído (MITCHELL, 1998).

### 2.1.4.2 - OPERADOR GENÉTICO DE MUTAÇÃO

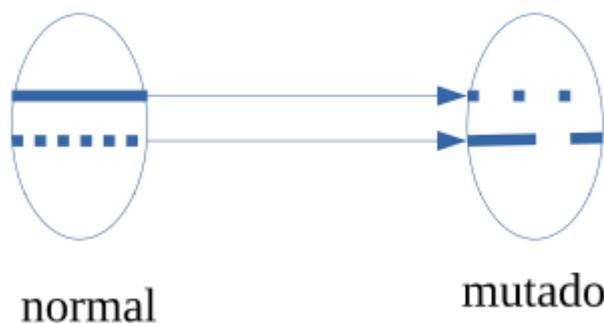
A mutação é vista como um operador unário, sendo aplicado a um genótipo e produzindo um descendente ligeiramente adaptado. Esse operador é sempre estocástico, ou seja, a sua saída irá depender de uma série de fatores aleatórios (EIBEN e SMITH, 2015).

A mutação garante que haja diversidade das características dos indivíduos na população, de tal forma, permite que ocorra introdução de informações que antes não estavam presentes em nenhum indivíduo da população. Proporcionando também, *exploration* no AG, oferecendo uma maior possibilidade de que mais pontos do espaço de busca sejam avaliados (GUIMARÃES, 2005).

De forma análoga a natureza, essa operação deve ser aplicada apenas em uma pequena parcela da população, senão pode perder suas características fundamentais e ter um comportamento de um algoritmo aleatório (*random walk*).

O operador genético de mutação pode ser representado pela figura 2.5, onde em um cromossomo  $x$  pode ocorrer a mutação dos genes desse cromossomo.

**Figura 2.5 - Operador Genético de Mutação**



**Fonte:** (Elaborado pelo Autor, 2022)

### 2.1.4.2.1 - MUTAÇÃO POR COMPLEMENTO

A mutação por complemento recebe um cromossomo, um conjunto de posições  $\{j_1, \dots, j_w\} \subseteq \{1, \dots, m\}$ , e um número randômico  $r_m \in [0, 100]$ , e retorna o cromossomo dado, se  $r_m$  é maior que a probabilidade de mutação  $prob_m$ , caso contrário, um novo cromossomo cuja informações são apresentadas nas posições  $j_1, \dots, j_w$ , tem seu valor original invertido.

A mutação por inversão de *bit* é uma função *flip* definida como:

$$flip(\vec{x}_1^m, \{j_1, \dots, j_w\}, r_1) = \vec{y}_1^m, \quad \text{onde} \quad \forall i \in \{1, \dots, m\} - \{j_1, \dots, j_w\} \vec{y}_i^i = \vec{x}_i^i \quad \text{e} \\ \forall i \in \{j_1, \dots, j_w\} ((\vec{x}_i^i = 0) \implies (\vec{y}_i^i = 1)) \wedge ((\vec{x}_i^i = 1) \implies (\vec{y}_i^i = 0)).$$

### 2.1.4.3 - SELEÇÃO

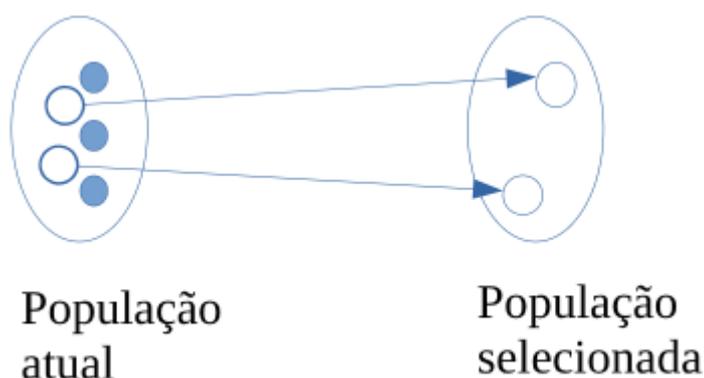
Esse operador satisfaz bem a teoria de Charles Darwin (1959), pois é através deles que os indivíduos mais bem adaptados ao ambiente sobrevivem e conseqüentemente existe uma maior probabilidade de passar o seu genes para os seus descendentes (NUNES, 2018). Os cromossomos são submetidos a uma função de adaptação, que determina uma pontuação ao indivíduo de acordo com as suas qualidades.

A probabilidade de um dado cromossomo ser selecionado para reproduzir suas características, está diretamente relacionada com a sua pontuação, ou seja, mesmo que exista a probabilidade de um cromossomo conseguir reproduzir seu genes, essa probabilidade pode se tornar inferior, se comparado com outro cromossomo bem avaliado (NUNES, 2018).

Esse processo não deve descartar os indivíduos pouco adaptados, isso porque, mesmo com a baixa adaptação ao ambiente, eles podem conter características boas e que devem ser passadas para as próximas gerações (ROSA e LUZ, 2009).

A seleção pode ser representada pela figura 2.6, onde em uma dada população com  $x$  cromossomos, apenas os que satisfazem uma determinada condição, serão selecionados para compor a nova população.

**Figura 2.6** - Operador Genético de Seleção



**Fonte:** (Elaborado pelo Autor, 2022)

#### 2.1.4.3.1 - SELEÇÃO ELITISTA

A função de seleção elitista recebe uma população  $P$  e um valor inteiro  $x$ , e retorna uma subpopulação  $s$  da população  $P$  formada pelos cromossomos  $x$  mais adaptados da população  $P$ .

A seleção elitista é definida por uma função *elite*, definida como:

$elite(\vec{p}_1^n, x) = \vec{s}_1^x$ , onde  $\forall i \in \{1, 2, \dots, n-1\} \forall j \in \{i, 2, \dots, n\} f(p_i) \geq f(p_j)$  e  $\forall i \in \{1, 2, \dots, x\} (s_i = p_i)$ .

## 2.1.5 - PARÂMETROS DOS ALGORITMOS GENÉTICOS

A escolha dos parâmetros ideais para um AG é considerado um problema no momento do desenvolvimento, entre eles, tamanho da população, taxa de mutação e taxa de cruzamento (BARCELLOS, 2000). Tais parâmetros, são essenciais tanto para o comportamento do AG quanto para o resultado esperado, onde se houver uma alta taxa de mutação, o algoritmo pode se tornar totalmente aleatório, caso seja baixa, pode impactar diretamente na diversidade da população (NUNES, 2018). A taxa de cruzamento elevada resulta em uma convergência rápida do AG, e caso seja muito baixa, pode dificultar a busca por soluções ótimas.

De acordo com Chebbi e Chaouachi (2015), existem dois tipos de parâmetros, os qualitativos e os quantitativos. Os parâmetros quantitativos são referentes aos diferentes tipos de taxas que podem ser utilizadas no AG, como taxas de mutação, de cruzamento. Já os parâmetros qualitativos são simbólicos, possuindo um domínio finito e não possuindo métrica de distância específica, com por exemplo, funções que definem os pontos de cortes (um, dois, três) no cruzamento.

O *Parameter tuning problem*<sup>3</sup> é dito como os melhores valores para as características de um AG (CHEBBI e CHAOUACHI, 2015). Os parâmetros podem ser constantes durante a execução ou pode ser modificado em tempo de execução, essas duas formas são denominadas de ajuste de parâmetros e controle de parâmetros (CHEBBI e CHAOUACHI, 2015). No ajuste de parâmetros, temos que os valores definidos antes de iniciar a execução do algoritmo permanecem constantes até o fim de todo o processo, enquanto que o controle de parâmetros, os valores são alterados durante a execução do programa (NUNES, 2018).

### 2.1.5.1 - TAXA DE CRUZAMENTO

Segundo Linden (2008) a taxa de cruzamento deve ser alta, variando entre 60% e 95%, ou seja, essa variação dependendo do quanto seu problema precisar de indivíduos com novas características. Entretanto, outros autores realizaram testes com taxa de cruzamento variando entre 11% até 90% (NUNES, 2018).

---

<sup>3</sup> Os valores dos parâmetros são especificados no início de um algoritmo. Os valores não mudam enquanto o algoritmo está em execução (CHEBBI e CHAOUACHI, 2015)

### 2.1.5.2 - TAXA DE MUTAÇÃO

De acordo com a literatura, a taxa de mutação deve variar entre 0,5% e 5%, caso o valor ultrapasse 5% torna a busca aleatória (CATARINA e BACH, 2003). Entretanto, existem estudos na literatura que realizaram testes com a taxa de mutação variando de 5% até 90% (NUNES, 2018).

## 2.2 - TRABALHO CORRELATOS

A ideia de preservar um *schema* que está presente na população nas próximas gerações já foi explorada por outros trabalhos, como pode ser visto em uma busca simples nas bases de dados: *IEEE Xplore*, e ACM, onde:

- No *IEEE Xplore* a busca resultou em um total de 36 documentos;
- Na ACM a busca retornou um total de 131 documentos.

Neste trabalho, o operador de cruzamento está diretamente ligado a dois propósitos. O primeiro é que, o operador de cruzamento realiza a recombinação dos blocos de construção, ou seja, combina os blocos que estão contidos em *strings* bem avaliadas na população em uma única *string*. Já o segundo propósito é que, esse operador auxilia na descoberta de novos blocos de construção, à medida que coloca alelos em novos contextos, para assim, realizar a descoberta de novos blocos (HOLLAND, 2000).

É evidente que dificilmente pode-se identificar que um *schema* é viável dentro do espaço de busca. Por conta disso, o artigo de Fan et al. (2005), apresenta um novo operador de cruzamento que auxilia na prevenção dos *schemata* que apresentam um desempenho promissor. Onde foi aplicado o conceito de redução de atributos na teoria de conjuntos aproximados (*Rough Sets*). Com essa aplicação, pode-se encontrar os genes chaves do cromossomo, pois estes genes irão distinguir se o cromossomo possui um desempenho promissor. Logo, é possível escolher os *schemata* determinados por esse gene chave como um candidato a ser avaliado. Se na avaliação provar que o *schema* é promissor para o problema, ele será preservado no cruzamento, adicionando assim algumas restrições sobre a escolha dos pontos de corte, de tal modo que os *schemata* possuam uma maior probabilidade de serem conservados durante o cruzamento.

Para auxiliar na eficácia do operador de cruzamento, Zhang et al. (2007) propôs um uma abordagem denominada de LCC (*looseness-controller crossover*). Nessa abordagem, é

utilizado uma pesquisa local de *hill-climbing* com o intuito de buscar bons blocos de construção, onde será introduzido um peso denominado de *looseness* que auxilia na identificação dos blocos de construções que existem na população. A utilização desse peso, favorece a preservação dos blocos de construção, pois acontece a escolha de pontos de cortes apropriados que preservem o *schema*.

Uma variação do problema do caixeiro viajante, é a dos múltiplos caixeiros viajantes. Para resolver essa variação, Yuan (2013) sugeriu um novo operador de cruzamento, denominado de cruzamento de cromossomo em duas partes (TCX), com o intuito de encontrar soluções quase ótimas. Para esse cruzamento, foi adotado a representação do cromossomo em duas partes, com o intuito de minimizar o tamanho do espaço de busca do problema.

Diante do exposto, nota-se que ainda há lacunas no campo de estudo deste trabalho, pois muitos autores propõem um novo operador de cruzamento de tal forma que analisam apenas os *schemas* que estão presentes na população. De tal forma, a preservação dos blocos de construções presentes entre os pais não foram devidamente explorados, com o intuito de melhorar a convergência e qualidade das soluções encontradas.

No próximo capítulo serão apresentados os conceitos, bem como o desenvolvimento do operador de cruzamento proposto neste trabalho. Tal operador, possui o intuito de aumentar a qualidade dos indivíduos, como também auxiliar na convergência do AG.

## CAPÍTULO 3

### UM NOVO MODELO DE OPERADOR DE CRUZAMENTO BASEADO NA PRESERVAÇÃO DE BLOCOS DE CONSTRUÇÃO

#### 3.1 - INTRODUÇÃO

De acordo com Holland (1992) o principal desafio de um sistema adaptativo é quando um conjunto de estrutura  $a$  é muito grande e as funções de adaptação  $\mu_E$  possui muitos pontos de máximos locais. Sendo de extrema importância, que esses sistemas busquem formas de ir contra a busca prolongada. Segundo Aizawa e Wah (1994), o tamanho da população possui influência direta no desempenho dos AG.

Uma forma de demonstrar teoricamente como um algoritmo genético consegue se desvencilhar dos ótimos locais é baseado no estudo da teorema do *schema*, proposto por Holland, para realizar a representação dos subespaços do espaço total de busca, onde os elementos são os indivíduos da população (CARVALHO, 2008).

Este trabalho irá utilizar a definição formal de *schema* do trabalho de Vilhena (2003).

**Definição 3.1: Schema** - É uma  $n$ -úpla  $s$ , tal que  $s = (s_1, \dots, s_m)$ , onde cada elemento  $s_j$  de  $s$  é uma variável  $v_1, v_2, \dots, v_w$  com  $w \leq m$ , ou sendo um constante de valor 0 ou 1.

De acordo com essa definição, dado um *schema*  $s = (s_1, \dots, s_m)$ , o subespaço de busca dado por  $s$  é denominado de  $\vec{s}$ , que será formado por todos os vetores  $\langle s_1, s_2, \dots, s_m \rangle$ . Esses vetores são obtidos a partir da substituição das variáveis existentes em  $s$  pelo valor das constantes 0 e 1. O conjunto dos *schemata* de uma população  $P$  é denominado por  $S^P$ .

**Definição 3.2: O espaço representado por um schema** - dado o schema  $s = (s_1, \dots, s_m)$ ,

o espaço que representa esse schema é o conjunto  $\vec{s}$ , esse vetor pode ser definido como:

$$\vec{s} = \{x = \langle x_1, x_2, \dots, x_m \rangle / \forall 1 \leq j \leq m (j \in \text{const}(s) \rightarrow x_j = s_j) \vee (j \in \text{var}(s) \rightarrow x_j = 0 \vee x_j = 1)\}$$

Na definição de  $\vec{s}$ , sabe-se que *const* é uma função que recebe o *schema* e retornar o conjunto de posições do *schema* que existe ocorrência de uma constante. E o *var* é uma função que recebe o *schema* e retornar o conjunto de posições onde há a ocorrência de uma variável.

Seja o schema  $s = (v_1, 0, v_2, 1, v_3)$ . Sabendo que o subespaço desse é *schema* dado por  $\vec{s} = \langle 0, 0, 0, 1, 0 \rangle, \langle 0, 0, 0, 1, 1 \rangle, \langle 1, 0, 0, 1, 0 \rangle, \langle 0, 0, 1, 1, 1 \rangle, \langle 1, 0, 0, 1, 1 \rangle, \langle 1, 0, 1, 1, 0 \rangle, \langle 1, 0, 1, 1, 1 \rangle$ . É fácil ver que  $\vec{s}$  é formado pelos vetores que apresentam os valores das constantes 0 e 1 nas posições definidas no *schema*  $s$ , onde  $\text{const}(s) = \{2, 4\}$ . Sabe-se também que,  $\vec{s}$  é formado por vetores que apresentam os valores 0 e 1 nas posições definidas no *schema*  $s$  como uma variável, onde  $\text{var}(s) = \{1, 3, 5\}$ .

O número de elementos que são representados no espaço de um *schema*  $s$  é dado através da ordem de  $2^w$ , onde  $w$  representa o número de variáveis do *schema*. Isso é possível pelo fato de todas as variáveis do *schema* assumirem os valores 0 ou 1. Por exemplo, no *schema*  $s = (v_1, 0, v_2, 1, v_3)$ , existe um total de três variáveis, logo, considera-se que  $w = 3$ . Diante disso, sabe-se que o número de elementos do espaço deste *schema* é de  $2^3 = 8$ .

Dado um *schema*  $s = (s_1, \dots, s_m)$  e um indivíduo  $\langle a_1, \dots, a_m \rangle$  da população, considera-se que esse indivíduo pertence ao subespaço representado por  $s$ , se e somente se,  $\forall 1 \leq j \leq m (j \in \text{const}(s) \rightarrow a_j = s_j)$ . Seja o *schema*  $s = (v_1, 0, v_2, 1, v_3)$  e o indivíduo  $a = \langle 1, 1, 0, 1, 0 \rangle$ , sabe-se que esse indivíduo não pertence ao *schema*  $s$ , pois o caractere da segunda posição do indivíduo é divergente do declarado na constante do *schema*  $s$ . Se considerarmos o indivíduo  $b = \langle 1, 1, 0, 1, 0 \rangle$  e o mesmo *schema*  $s$ , percebe-se que esse novo

indivíduo pertence ao *schema*. Pois, todos os caracteres do indivíduo  $b$  são iguais aos que foram declarado como constante no *schema*  $s$ .

Para qualquer *schema*  $s$  de uma população  $P$ , existe um valor de adaptação que é associado ao *schema*  $s$ . Esse valor serve para mostrar o quanto o subespaço é relevante para a resolução do problema em questão.

**Definição 3.3: Adaptação do *schema*** - a adaptação do *schema* é definida por uma função  $\Phi$ , tal que  $\Phi: S^p \times P \rightarrow IR$

$$\Phi(s, P) = \frac{\sum_{x \in X} f(x)}{\text{card}(X)}$$

Nessa função  $\Phi$ ,  $X$  representa o conjunto dos indivíduos da população  $P$  que pertencem ao espaço representado pelo *schema*  $s$ . Sabe-se ainda que  $s$  representa uma função de adaptação. E  $\text{card}(X)$  representa uma função que, dado um conjunto finito, retorna a cardinalidade desse conjunto.

Se a representação dos subespaços de um algoritmo e seu grau de adaptação forem conhecidos, Holland demonstrou que, através do comportamento probabilístico deste algoritmo, os subespaços de uma população na interação  $i$  são menos adaptados que os da interação  $i + 1$ .

Sabe-se que o primeiro operador aplicado em uma população  $P$  é a seleção. Esse operador é responsável por indicar os *schemata* a serem investigados, isso ocorre a partir do armazenamento dos indivíduos da população  $P$  que pertencem ao *schemata* na população intermediária  $P'$ . O número de indivíduos da população  $P'$  que pertence ao *schema* da população  $P$  é dado pela equação abaixo:

$$\xi(s, P') = \xi(s, P) \times h \times \frac{\Phi(s, P)}{\sum_{x \in X} f(x)}$$

Onde  $\xi(s, P')$  é o número de indivíduos que pertencem ao *schema*  $s$  da população  $P$ , o tamanho da população  $P'$  é representado por  $h$ , enquanto que a adaptação do *schema*  $s$  na população  $P$  é dado por  $\Phi(s, P)$ . Já  $f$  é uma função de adaptação definida no tópico 2.1.2. Enquanto  $X$  é o conjunto dos indivíduos da população  $P$ .

A quantidade de indivíduos que compõem um *schema*  $s$  de uma população em construção  $P$  é igual ao número de indivíduos que pertencem ao *schema*  $s$  na população  $n - k$  indivíduos mais adaptados da população atual  $P$ , acrescentando ainda, o número de indivíduos do *schema*  $s$  na população dos  $k$  indivíduos mais adaptados da população  $P''$ , como pode ser observado na equação a seguir:

$$\xi(s, P') = \xi(s, P) \times (n - k) \times \frac{\phi(s, P)}{f(P)} + \xi(s, P'') \times k \times \frac{\phi(s, P'')}{f(P'')}$$

A quantidade de indivíduos do *schema*  $s$  da população  $P''$  é igual ao número de indivíduos que pertencem ao *schema*  $s$  de  $P'$  multiplicado pela probabilidade desse *schema* ser preservado pela ação dos operadores genéticos de cruzamento  $p_{\Delta}$ , mutação  $p_{\Omega}$  e inversão  $p_{\nabla}$ , como pode ser visualizado abaixo.

$$\xi(s, P'') \geq \xi(s, P') \times p_{\Delta} \times p_{\Omega} \times p_{\nabla}$$

A probabilidade que um *schema*  $s$  tem de ser preservado pela ação do operador de cruzamento é:

$$p_{\Delta} \geq 1 - p_c \times \frac{\Delta}{m-1}$$

Onde  $\Delta$  se refere a distância entre a primeira e última ocorrência de uma constante no *schema*  $s$ . O sinal maior igual, é adotado para justificar que se  $s \in S_a \cup S_b$ , então todos os indivíduos gerados pertencem ao *schema*  $s$ . Enquanto  $p_c$  representa a probabilidade de cruzamento.

O termo  $\frac{\Delta}{m-1}$  retrata a única possibilidade de cruzamento entre indivíduos que pertencem a *schemata* distintos. Onde, sejam os indivíduos  $\langle a_1, a_2, \dots, a_m \rangle$  e  $\langle b_1, b_2, \dots, b_m \rangle$  que pertencem a dois conjuntos de *schemata* distintos  $S_a$  e  $S_b$ , onde  $S_a \cap S_b = \emptyset$  e  $s \in S_a \cup S_b$ . Para preservar um *schema*  $s$  nessa situação, é necessário que os pontos de cortes de valor  $z$  não coincida com uma das posições entre a primeira e a última ocorrência de uma constante do *schema*  $s$ .

Por exemplo, seja  $p_c = 0,55$  a probabilidade de preservar o *schema*  $s = (v_1, v_2, 0, v_3, v_4, 0, 1)$  pela ação do operador de cruzamento nos indivíduos  $a = (0, 1, 1, 1, 0, 1, 0, 1)$  e  $b = (0, 1, 1, 0, 1, 0, 1, 0)$ , que pertencem ao conjunto dos *schemata*  $((v_1, v_2, 1, 1, v_3, v_4, v_5, v_6), (0, 1, 1, v_1, v_2, v_3, v_4, v_5), (v_1, 1, v_2, v_3, 0, 1, v_4, v_5), (v_1, v_2, 1, v_3, v_4, v_5, v_6, v_7), (v_1, 1, v_2, v_3, v_4, v_5, v_6, v_7))$  e  $((0, 1, 1, v_1, v_2, v_3, v_4, v_5), (v_1, v_2, 1, v_3, v_4, v_5, v_6, v_7), (v_1, 1, v_2, v_3, v_4, v_5, v_6, v_7), (v_1, 1, v_2, 0, v_3, v_4, 1, 0))$ , temos que:

$$p_{\Delta} \geq 1 - 0,55 \times \frac{8-2}{8-1} = p_{\Delta} \geq 0,52$$

Uma fórmula semelhante que define a probabilidade do *schema*  $s$  ser preservado pela ação do operador de mutação, pode ser visualizada abaixo:

$$p_{\Omega} = (1 - p_m)^{m-w}$$

Sabe-se que  $p_m$  é a probabilidade de mutação, enquanto  $m$  e  $w$  estão relacionados ao tamanho e número de variáveis do *schema*  $s$  respectivamente.

A partir dessa equação é fácil vê que, a única possibilidade de mutar um indivíduo  $\langle a_1, a_2, \dots, a_m \rangle$  que pertence ao conjunto de um *schema*  $S_a$ , tal que  $s \in S_a$ . Para preservar o *schema*  $s$ , é preciso fazer com que os pontos de mutação deste indivíduo não coincidam com as posições das constantes do *schema*.

Por exemplo, seja  $p_m = 0,85$  a probabilidade de preservar o *schema*  $s = (v_1, 1, v_2, 0, v_3, v_4, 1, 0)$  pela ação do operador de mutação no indivíduo  $a = (0, 1, 1, 0, 1, 0, 1, 0)$ , que pertence ao conjunto de *schema*  $((0, 1, 1, v_1, v_2, v_3, v_4, v_5), (v_1, v_2, 1, v_3, v_4, v_5, v_6, v_7), (v_1, 1, v_2, v_3, v_4, v_5, v_6, v_7), (v_1, 1, v_2, 0, v_3, v_4, 1, 0))$ . De tal forma, a probabilidade de preservação do *schema*  $s$  pelo operador de mutação é dado por:

$$p_{\Omega} = (1 - 0,85)^{8-4} = 0,0005$$

A probabilidade do *schema*  $s$  ser preservado pela ação do operador de inversão segundo é dado pela inequação abaixo.

$$p_{\nabla} \geq 1 - p_i \times \frac{m-w}{w}$$

Nessa inequação  $p_i$  é a probabilidade de inversão, o tamanho do *schema* é dado por  $m$ , e o número de variáveis do *schema*  $s$  é dado por  $w$ .

Através dessa inequação, se existir um indivíduo  $\langle a_1, a_2, \dots, a_m \rangle$ , que pertence ao conjunto do *schema*  $S_a$ , tal que  $s \in S_a$ . A única possibilidade de aplicar a inversão sobre esse indivíduo é se o *schema*  $s$  for conservado. O *schema*  $s$  poderá ser conservador se e somente se, as posições do indivíduo que terão seus valores alterados não coincidam com aquelas das constantes do *schema*  $s$ .

Por exemplo, seja o *schema*  $s = (v_1, 1, v_2, 0, v_3, v_4, 1, 0)$ , o indivíduo  $a = (0, 1, 1, 0, 1, 0, 1, 0)$  que pertence ao conjunto do *schema*  $((0, 1, 1, v_1, v_2, v_3, v_4, v_5), (v_1, v_2, 1, v_3, v_4, v_5, v_6, v_7), (v_1, 1, v_2, v_3, v_4, v_5, v_6, v_7), (v_1, 1, v_2, 0, v_3, v_4, 1, 0))$  e  $p_i = 0,95$ . A probabilidade de preservação do *schema*  $s$  pela ação do operador genético de inversão sobre o indivíduo  $a$  é de:

$$p_{\nabla} \geq 1 - 0,95 \times \frac{8-4}{4} = 0,05$$

Ao substituir as inequações da probabilidade de preservação de um *schema* pela ação das operações de cruzamento e inversão, e também a equação da probabilidade de preservação de um *schema* pela ação do operador de mutação na equação do número de indivíduos que pertencem a um *schema*  $s$  da população  $P''$ , tem-se a seguinte expressão:

$$\xi(s, P'') \geq \xi(s, P') \times \left(1 - p_c \times \frac{\Delta}{m-1}\right) \times (1 - p_m)^{m-w} \times \left(1 - p_i \times \frac{m-w}{w}\right)$$

Ao aplicar essa expressão à equação do número de indivíduos que pertencem ao *schema*  $s$  da população  $P$ , obtém-se a seguinte expressão:

$$\xi(s, P) \geq \xi(s, P') \times \left[ (n - k) \times \frac{\Phi(s, P)}{f(P)} + (1 - p_c \times \frac{\Delta}{m-1}) \times (1 - p_m)^{m-w} \times (1 - p_i \times \frac{m-w}{w}) \times k \times \frac{\Phi(s, P'')}{f(P'')} \right]$$

Através dessa nova inequação, Holland provou que seu algoritmo preserva os *schemata* mais adaptados que surgem durante a execução do algoritmo. Ou seja, isso significa

que o algoritmo  $R$  realmente vai de encontro aos subespaços formados pelos *schemata* mais adaptados das populações anteriores a atual.

A teoria dos *schemata* é mostrada como o algoritmo genético de Holland, que realiza a sobreposição dos *schemata* investigados por outros *schemata* mais adaptados em uma iteração, mas não demonstra como este algoritmo chega a população formada somente por cromossomos adaptados.

A convergência do algoritmo genético de Holland foi tratada por Goldberg (1989) sob o nome de hipótese do bloco de construção, segundo a qual, um algoritmo genético busca cromossomos com um desempenho ótimo pela justaposição de *schemata* pequenos, de baixa ordem e com alta adaptação, denominados de blocos de construção.

A hipótese do bloco de construção diz que: Um algoritmo genético converge para uma população formada por cromossomos pertencentes aos *schemata* de tamanho pequeno, de baixa ordem e com adaptação acima da média dos *schemata* existentes na população atual.

Como o próprio Goldberg (1989) observa, para que a hipótese do bloco de construção seja bem sucedida é necessário que as informações codificadas no cromossomo não sejam dependentes entre si, já que este tipo de relacionamento não pode ser capturado pelo *schema*. A presença de dependência entre as informações codificadas no cromossomo (*epistasis*) engana o algoritmo genético, fazendo o algoritmo convergir para um bloco de construção que contém parte destas informações (GOLDBERG, 1989).

O bloco de construção é dado pela função *block*, que é definida como:  $block(\vec{x}_1^m, \vec{y}_1^m, J, r_c) = (\vec{z}_1^m, \vec{w}_1^m)$ , onde  $J = \{1, \dots, m\}$  é o ponto de corte,  $r_c \in [0, 100]$  é um número randômico,  $P\vec{a}_1^m$  é a população formada somente com o cromossomo  $\vec{a}_1^m$ ,  $S^{pz_1^m} = S^{p\vec{x}_1^m} \cup S^{p\vec{y}_1^m}$  e  $S^{pw_1^m} = S^{p\vec{x}_1^m} \cup S^{p\vec{y}_1^m}$ .

A hipótese dos blocos de construção é de extrema importância para os AG, pois não é apenas importante aplicar os operadores genéticos de tal forma que preserve os blocos de construção, como também, criar representações de vetores em que os blocos de construção possuam importância para o indivíduo (KAZADI, 1998).

### 3.3 - CRUZAMENTO PROPOSTO

O operador de cruzamento proposto neste trabalho baseia-se na preservação não dos *schemata* presentes em todos os indivíduos promissores da população, mais na preservação de

blocos de construção existentes entre os pais, que é um subconjunto dos blocos de construção presentes na população. Esta alteração torna o algoritmo mais rápido e adequado à teoria da Evolução das Espécies.

Ao ser escolhido um conjunto de pais para gerar descendentes, eles são analisados para identificar os blocos de construção presentes em todos, baseando-se nessa lista dos blocos de construção presentes nos indivíduos pais é gerada uma lista com os pontos de corte válidos. Essa lista de pontos de corte que foi definida, retorna para a função de cruzamento, e a partir desse momento, é selecionado de forma randômica uma posição que está dentro da lista de pontos de corte.

Foi realizada a implementação do cruzamento de um ponto de corte e de dois pontos de cortes, com o intuito de analisar qual o tipo de cruzamento é mais propício a auxiliar na convergência do algoritmo.

No capítulo seguinte será abordado sobre as técnicas que foram utilizadas para desenvolver os experimentos deste trabalho, bem como, o cruzamento proposto, as especificações dos algoritmos desenvolvidos, o ambiente de teste e as especificações do ambiente.

## CAPÍTULO 4

### MATERIAIS E RESULTADOS

#### 4.1 - MATERIAIS E MÉTODOS

##### 4.1.1 - AMBIENTE DE TESTE

Os algoritmos foram desenvolvidos e testados no *Google Collaboratory*, nas seguintes especificações (GOOGLE, 2017):

- Memória RAM:  $\approx 12\text{ GB}$
- Disco:  $\approx 108\text{ GB}$

##### 4.1.2 - ALGORITMO GENÉTICO DESENVOLVIDO

###### 4.1.2.1 - ESPECIFICAÇÕES GERAIS

O algoritmo genético foi desenvolvido na linguagem de programação *Python*<sup>4</sup>, na versão 3.7.17.

Para auxiliar no desenvolvimento do algoritmo, foi necessário a utilização das bibliotecas *numpy*, *math*, *statistics*, *random* e *time*.

---

<sup>4</sup> <https://www.python.org/doc/>

### 4.1.2.2 - ESPECIFICAÇÃO DOS ALGORITMOS

Para realizar os experimentos foi necessário realizar a implementação do algoritmo genético de Holland e do algoritmo genético com o cruzamento proposto (3.2).

Entretanto, como foi proposto realizar esses experimentos em dois problemas de otimização, sendo um funcional e o outro combinatorial, foi necessário desenvolver os algoritmos com configurações distintas. Diante disso, foi desenvolvido duas versões para o algoritmo de Holland e duas para o algoritmo genético com o cruzamento proposto (3.2).

### 4.1.3 - MÉTRICAS DE AVALIAÇÃO

Os algoritmos foram executados por 100 iterações, com o intuito de mensurar a convergência dos algoritmos genéticos, bem como a qualidade dos cromossomos encontrados, sendo esses, fatores cruciais na para os algoritmos genéticos.

O algoritmo de Holland e o algoritmo genético com o operador proposto neste trabalho (3.2), foram executados com as seguintes combinações dos parâmetros:

**Quadro 4.1** - Combinação dos parâmetros para o algoritmo genético de Holland e algoritmo genético com o operador de cruzamento proposto

Tamanho População	Tamanho Cromossomo	Taxa de cruzamento (%)	Taxa de mutação(%)
30	10	60	10
30	50	70	10
30	100	90	10

**Fonte:** (Elaborado pelo Autor, 2022)

Ambos os algoritmos são submetidos aos mesmos parâmetros, com o intuito de manter as mesmas condições de testes para obter o resultado. Pode-se notar que foi optado por deixar fixa a taxa de mutação, isso se deu pelo fato do artefato estudado neste trabalho ser o operador de cruzamento.

Ao fim de cada execução do quadro 4.1, foi possível realizar a plotagem dos gráficos com os valores extraídos de cada execução. Esses gráficos são compostos pelo eixos x e y, onde no eixo x, está presente a quantidade de gerações que o AG obteve. Já no eixo y, encontra-se a qualidade dos indivíduos.

Os gráficos escolhidos para realizar a análise dos resultados dessa pesquisa, foram os gráficos em linha, onde, em cada gráfico será possível notar 3 linhas (verde, azul e vermelha). As linhas de cor verde, representam os algoritmos que foram executados com a taxa de cruzamento de 60%. As linhas de cor azul representam os algoritmos com uma taxa de cruzamento de 70%. Enquanto que as linhas vermelhas, representam os algoritmos com taxa de cruzamento de 90%.

## 4.1.4 - PROBLEMAS PROPOSTO

### 4.1.4.1 - ONEMAX

O problema OneMax é muito utilizado para realizar testes em AG. Esse problema consiste em maximizar o número de uns em uma *string* binária de tamanho  $l$  (SOARES, 2014), sendo representado pela equação abaixo.

$$f_{UM}(X) = \sum_{i=0}^{l-1} x_i$$

Esse problema possui o caso genérico, onde busca-se minimizar o somatório de *bits* diferentes entre uma solução  $X$  e um vetor alvo  $O$  (SOARES, 2014). Essa generalização é representada por:

$$f_{UM}(X, O) = \sum_{i=0}^{l-1} |x_i - o_i|$$

De acordo com Goldberg (2002), o OneMax é utilizado para testar a eficiência dos algoritmos genéticos, isso ocorre no momento em que a escala do problema é aumentada, ou seja, o número de  $l$  *bits* da *string* é aumentado.

O problema OneMax é na maioria dos casos um ponto de partida interessante para realizar a verificação empírica. Onde, se uma equação falha em cenário relativamente “fácil”, provavelmente, ela não irá servir em situação com *log* mais complexo (GOLDBERG, 2002).

Para resolver esse problema, o algoritmo genético foi modelado nas seguintes especificações:

**Tamanho do cromossomo:** Os cromossomos variaram entre os tamanhos 10, 30 e 50.

**Tamanho da população:** Foi adotado o total de 30 indivíduos;

**Representação:** A representação mais adequada segundo Giguere e Goldberg (1998) é a binária, ou seja, composto apenas por zeros e uns;

**Função de adaptação:** Maior quantidade de *bits* 1 no cromossomo;

**Condição de Parada:** Se a adaptação do cromossomo for igual ao tamanho do cromossomo ou se uma adaptação  $z$  permanecer como a maior por 5 vezes consecutivas.

**Mutação:** Em ambos os algoritmos foi por complemento;

**Cruzamento:**

- Para o algoritmo genético de Holland foi aplicado apenas o cruzamento de um ponto de corte.
- Enquanto que no algoritmo genético com o teorema dos *schemata*, foi aplicado o cruzamento proposto nesse artigo (3.2), de duas formas, o de um ponto de corte e o de dois pontos de corte.

**Taxa de Mutação:** A taxa de mutação adotada em ambos os algoritmos é de 10%;

**Taxa de Cruzamento:** Foi definido um valor variando entre 60%, 70% e 90%, como visto no tópico (2.2.4.1)

#### 4.1.4.2 - CAIXEIRO VIAJANTE

O caixeiro viajante é um problema de otimização combinatória, fazendo parte da classes de problemas *NP-Hard*, que são os problemas que não podem ser resolvidos por um algoritmo polinomial (GUEDES; LEITE; ALOISE, 2005). Esse problema é estabelecido da seguinte forma:

- Dado um conjunto  $x$  de cidades e o custo entre elas, deve-se procurar o caminho mais barato para visitar todas as cidades do caminho uma única vez e retornar para a cidade de origem.

De acordo com Enima (2019), esse problema pode ser modelado matematicamente, da seguinte forma:

$$\text{Minimizar } z = \sum_{j=1}^n \sum_{i=1}^n c_{ij} x_{ij}$$

Sujeito a:

$$\sum_{i=1}^n x_{ij} = 1, \forall j \in N$$

$$\sum_{j=1}^n x_{ij} = 1, \forall i \in N$$

$$\sum_{i,j \in S} x_{ij} < |S| - 1, \forall S \subset N$$

$$x_{ij} \in \{0, 1\} \forall i, j \in N$$

Logo, temos que:

$N$ : É o número de nós da rede;

$\sum_{i=1}^n x_{ij} = 1, \forall j \in N$ : Todo fluxo que chega ao nó  $j$  é igual a 1;

$\sum_{j=1}^n x_{ij} = 1, \forall i \in N$ : Todo fluxo que sai do nó  $i$  é igual a 1;

$S$ : É um subgrafo de  $G$ ;

$|S|$ : É o número de vértices do subgrafo  $S$ ;

$\sum_{i,j \in S} x_{ij} < |S| - 1, \forall S \subset N$ : Devem ser evitados subciclos;

$$x_{ij} = \begin{cases} 1, & \text{se o fluxo passar pelo arco } (i, j) \\ 0, & \text{caso contrário} \end{cases}$$

O problema do caixeiro viajante, foi modelado seguindo as seguintes especificações:

**Tamanho do cromossomo:** Os cromossomos variaram entre os tamanhos 10, 30 e 50.

**Tamanho da população:** Foi adotado o total de 30 indivíduos;

**Representação:** A representação mais adequada para os cromossomos foi através dos números reais;

**Função de adaptação:** Menor caminho entre a cidade de início e destino, passando apenas uma vez por cada cidade;

**Condição de Parada:** Se o número  $y$  de gerações for atingido ou se uma adaptação  $z$  permanecer como a menor por 5 vezes consecutivas.

**Mutação:** Em ambos os algoritmos foi por complemento;

**Cruzamento:**

- Para o algoritmo genético de Holland foi aplicado apenas o cruzamento de um ponto de corte.

- Enquanto que no algoritmo genético com o teorema dos *schemata*, foi aplicado o cruzamento proposto nesse artigo (3.2), de duas formas, o de um ponto de corte e o de dois pontos de corte.

**Taxa de Mutação:** A taxa de mutação adotada em ambos os algoritmos é de 10%;

**Taxa de Cruzamento:** Foi definido um valor variando entre 60%, 70% e 90%, como visto no tópico (2.2.4.1)

## 4.2 - RESULTADOS E DISCUSSÃO

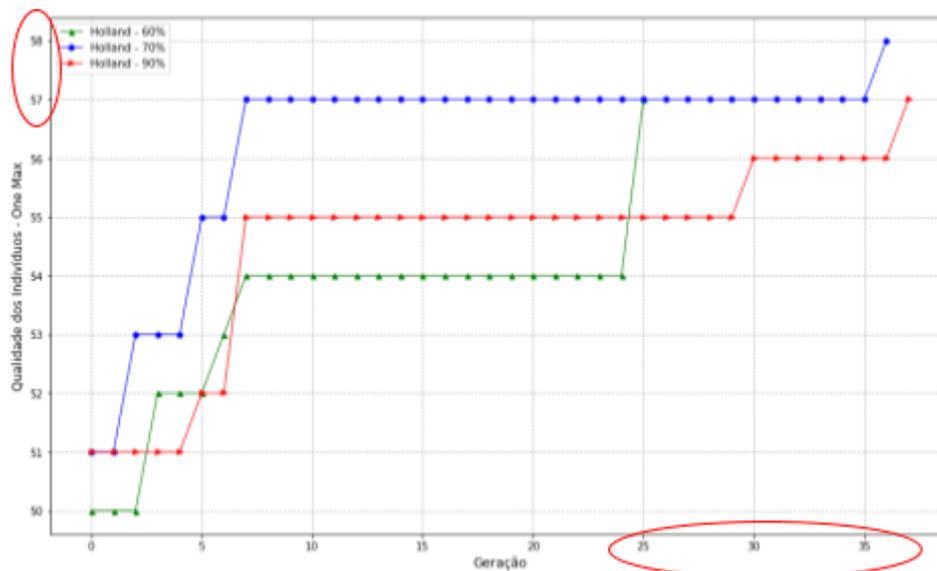
### 4.2.1 PROBLEMA ONEMAX

O problema do *OneMax* foi solucionado através da versão alfa (4.1.2.2.1), e ao final obteve-se um total de 9 gráficos, sendo divididos da seguinte forma:

- 3 para o algoritmo genético de Holland;
- 6 para o algoritmo genético com o operador de cruzamento proposto, sendo 3 o cruzamento de um ponto de corte e 3 para o de dois pontos de cortes;

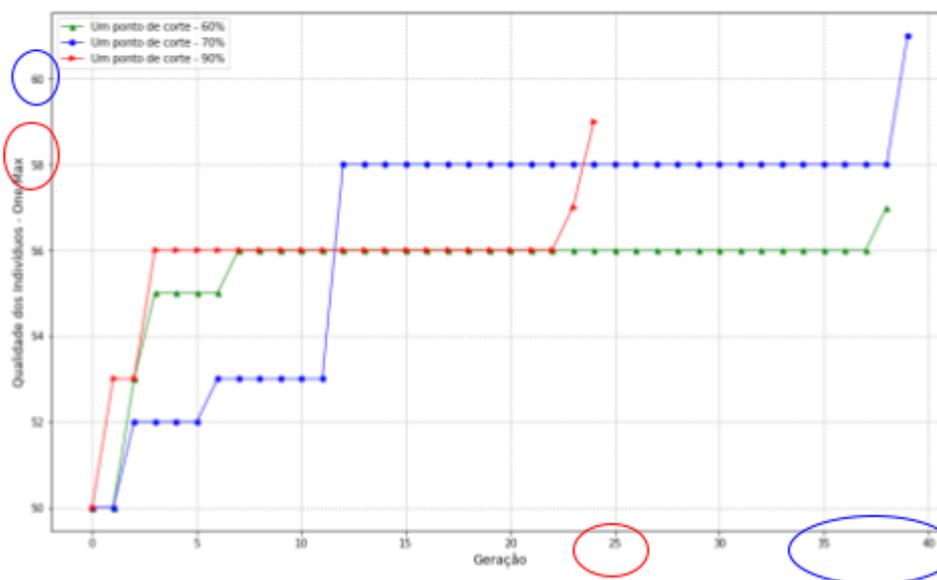
A seguir, pode-se visualizar os gráficos com os resultados obtidos para os algoritmos na seguinte combinação: população com tamanho trinta, cromossomo de tamanho cem, taxa de cruzamento variando entre 60%, 70% e 90%, e taxa de mutação a 10%. Os gráficos com os resultados das as outras combinações do quadro 4.1, podem ser encontrados no apêndice (J a K).

**Gráfico 4.1 - Resultado AG de Holland para OneMax**

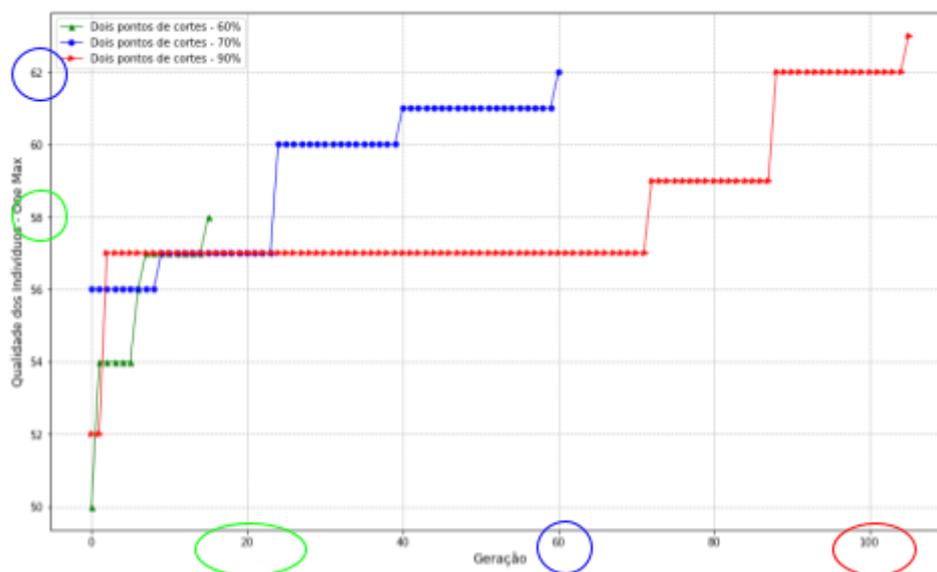


**Fonte:** (Elaborado pelo Autor, 2022)

**Gráfico 4.2- Resultado cruzamento proposto com um ponto de corte para OneMax**



**Fonte:** (Elaborado pelo Autor, 2022)

**Gráfico 4.3-** Resultado cruzamento proposto com dois pontos de cortes para OneMax

**Fonte:** (Elaborado pelo Autor, 2022)

No gráfico 4.1 pode-se notar que para o algoritmo genético de Holland, todas as 3 taxas convergiram antes de 40 gerações, onde a taxa de cruzamento a 60% convergiu em 25 gerações e conseguiu gerar indivíduos com qualidade 57. Já com a taxa de cruzamento a 70%, foi obtido o melhor desempenho na qualidade dos indivíduos se for comparado com as outras duas taxas, apresentando indivíduos com qualidade 58, e convergindo entre 35 e 40 gerações. Com a taxa de cruzamento de 90%, a convergência também ocorreu entre 35 e 40 gerações, entretanto gerou indivíduos de qualidade inferior à taxa de 70%.

No gráfico 4.2 nota-se que a taxa de cruzamento à 90%, o algoritmo consegue convergir antes de 25 gerações, gerando indivíduos com qualidade superior a 58, ou seja, ao realizar uma comparação com os dados do gráfico 4.1, consegue-se notar uma diferença no desempenho, pois ocorreu a geração de indivíduos de qualidade superior, bem como a convergência em menos gerações. A taxa de cruzamento definida a 70%, o algoritmo realiza a convergência entre 35 e 40 gerações, alcançando o melhor desempenho entre as três taxas do gráfico na qualidade dos resultados, onde gerou indivíduos com qualidade superior a 60. A taxa de cruzamento a 60% não gerou indivíduos com qualidade entre 56 e 58, ou seja, indivíduos inferiores aos gerados pelas taxas de cruzamento de 70% e 90%.

No gráfico 4.3 nota-se que o resultado com a taxa de cruzamento a 60%, consegue causar a convergência no algoritmo genético antes de 20 gerações, como também gerou indivíduos com a qualidade de 58, ou seja, se compararmos com o algoritmo de Holland do

gráfico 4.1, nota-se que foi alcançado a mesma qualidade, só que com aproximadamente 15 gerações a menos. A taxa de cruzamento a 70% gerou indivíduos com a qualidade superior a 62 e convergiu em 60 gerações. Já com a taxa a 90%, foram obtidas as melhores qualidades dos indivíduos entre os 3 gráficos (4.1, 4.2 e 4.3), onde os indivíduos foram superiores a 62, entretanto convergiu após 100 gerações, se tornando a combinação que obteve a convergência mais lenta.

**Quadro 4.2** - Soluções encontradas para o problema *OneMax*

Soluções para o problema <i>OneMax</i>						
	Algoritmo genético de Holland		Algoritmo genético com o operador proposto - Um ponto de Corte		Algoritmo genético com o operador proposto - Dois pontos de Corte	
Taxas de Cruzamento	Qualidade dos indivíduos	Gerações	Qualidade dos indivíduos	Gerações	Qualidade dos indivíduos	Gerações
60%	57	25	57	38	58	15
70%	58	36	61	39	62	60
90%	57	37	59	24	105	63

**Fonte:** (Elaborado pelo Autor, 2022)

No quadro 4.2 é possível observar de forma resumida os resultados expostos nos gráficos 4.1, 4.2 e 4.3. Sendo possível analisar que para a taxa de cruzamento de 60%, o melhor resultado obtido foi através do algoritmo genético com o operador proposto com dois pontos de cortes, onde foi obtido a maior qualidade de indivíduos em um menor número de gerações.

Para a taxa de cruzamento a 70%, observa-se no quadro 4.2 que, os algoritmos genéticos com um ponto de corte e dois pontos de corte, a diferença na qualidade foi de apenas um (1) ponto, entretanto nas gerações a diferença foi de vinte e um (21) pontos. De tal forma o algoritmo de um ponto de corte se torna viável em relação ao de dois pontos. Ao analisar o algoritmo de um ponto de corte e o algoritmo de Holland, é possível notar que entre ambos existe uma diferença de três (3) pontos na qualidade e nas gerações. Logo, o algoritmo de holland consegue satisfazer melhor as condições de maior qualidade em menor quantidade de gerações

Com a taxa de cruzamento fixada à 90%, é possível visualizar que os indivíduos de melhor qualidade foram alcançados através do algoritmo com dois pontos de cortes, entretanto, esse algoritmo também obteve a maior quantidade de gerações entre os três

algoritmos. Logo, tem-se que resultado mais satisfatório foi obtido do algoritmo proposto com um ponto de corte, gerando indivíduos com boa qualidade e no menor número de gerações. Ao comparar ambos os algoritmos, percebe-se um aumento de 77% na qualidade dos indivíduos e de 163% em relação às gerações. Enquanto que o algoritmo de holland, gerou qualidade de indivíduos inferiores ao algoritmo de um ponto de corte, bem como, convergiu em uma maior quantidade de gerações.

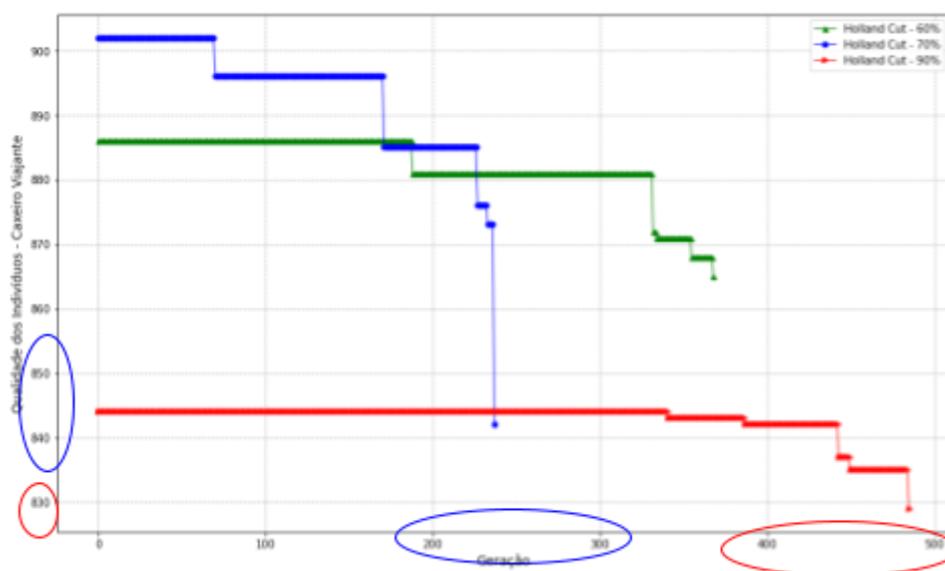
Logo, é notório que para o problema OneMax o algoritmo que melhor satisfaz as condições de qualidade de indivíduos e convergência do AG, foi o cruzamento proposto neste trabalho (3.2), com um ponto de corte, com taxa de cruzamento de 90%.

## **4.2.2 - PROBLEMA DO CAIXEIRO VIAJANTE**

O problema do caixeiro viajante foi resolvido com a versão beta (4.1.2.2.2), e ao fim da execução foi gerado um total de 9 gráficos, sendo divididos de forma análoga à seção anterior (4.2.1).

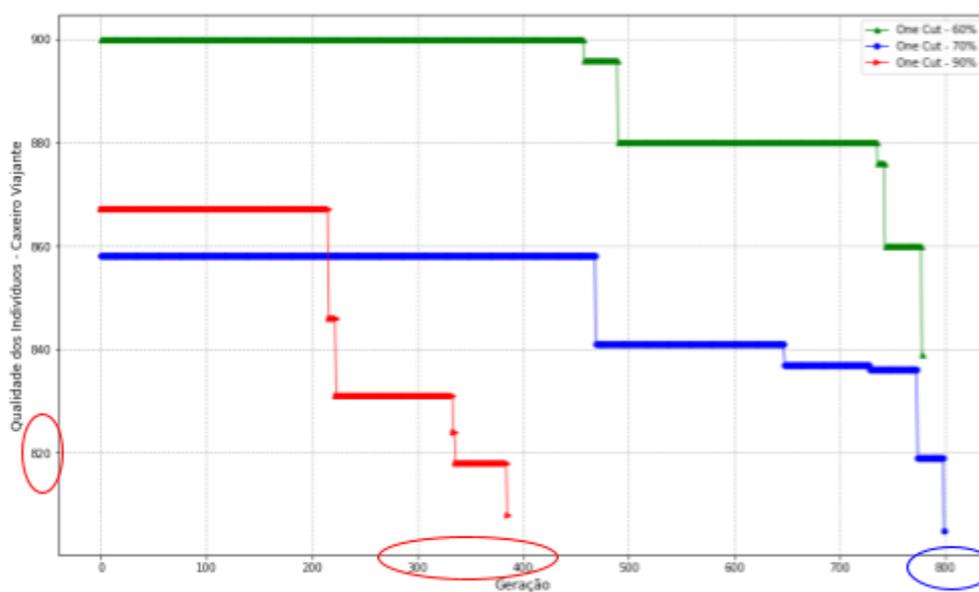
Esse problema foi submetido aos mesmos parâmetros que foram definidos na seção anterior (4.2.1), com tais valores: população de tamanho trinta, cromossomo de tamanho cem, taxa de de cruzamento variando entre 60%, 70% e 90%, e taxa de mutação a 10%. Os resultados obtidos por meio das outras combinações do quadro 4.1, podem ser encontrados no apêndice (A a F).

**Gráfico 4.4-** Resultado AG de Holland para o Caixeiro Viajante



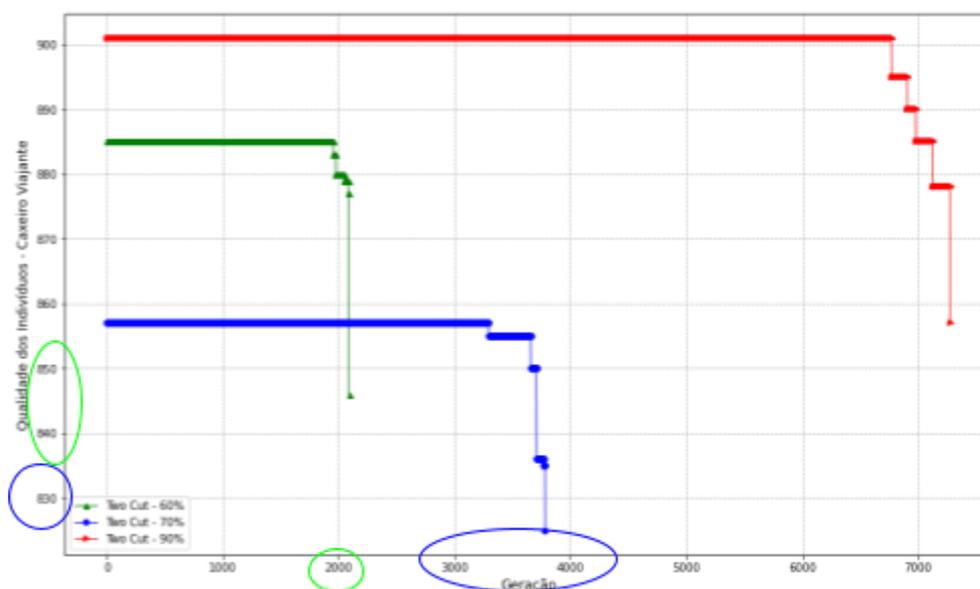
**Fonte:** (Elaborado pelo Autor, 2022)

**Gráfico 4.5-** Resultado cruzamento proposto com um ponto de corte para o Caixeiro Viajante



**Fonte:** (Elaborado pelo Autor, 2022)

**Gráfico 4.6-** Resultado cruzamento proposto com dois pontos de cortes para o Caixeiro Viajante



**Fonte:** (Elaborado pelo Autor, 2022)

O caixeiro viajante diferente do *OneMax* ele busca o menor caminho entre as cidades, ou seja, a qualidade dos indivíduos é sobre a minimização, e não sobre a maximização (como o *OneMax*).

No gráfico 4.4 pode-se observar os resultados obtidos pelo algoritmo genético de Holland. Onde, com a taxa de cruzamento de 70% a convergência do AG ocorreu entre 200 e 300 gerações, e a qualidade dos indivíduos ficou entre 850 e 840. Essa taxa foi a que apresentou melhor resultado no AG de Holland para o caixeiro viajante. Porém, ao observar a taxa de 90%, é possível notar que ele inicia com indivíduos de qualidade satisfatória entre 850 e 840, e consegue melhorar essa qualidade, onde os indivíduos ficam abaixo de 830. Contudo, essa taxa demora a realizar a convergência, convergindo próximo de 500 gerações. Ao definir a taxa de cruzamento em 60%, observou-se que a convergência do AG ficou entre 300 e 400 gerações, no entanto os indivíduos não obtiveram qualidade satisfatória se comparado com as outras duas taxas.

Nota-se que no gráfico 4.5 estão presentes os resultados obtidos para o algoritmo genético com o operador proposto de um ponto de corte. Ao definir a taxa de cruzamento a 90%, percebe-se que a convergência do AG ficou entre 300 e 400 gerações e gerou indivíduos

com qualidade entre 820 e 800. Enquanto que as taxas de cruzamento de 60% e de 70% convergiram entre 700 e 800 gerações. Onde, na taxa a 70% foram gerados indivíduos na mesma faixa que a taxa de cruzamento de 90%. Já para a taxa a 60%, foram gerados indivíduos com qualidade entre 840 e 820, não sendo indivíduos interessantes se comparados aos indivíduos gerados pelas duas outras taxas.

Primeira no gráfico 4.6 pode-se observar que a convergência foi a maior entre os três gráficos (4.4, 4.5 e 4.6), de tal modo que, em nenhum gráfico dos citados obteve-se um valor maior ou igual a 1000 gerações. A taxa de cruzamento que obteve a melhor convergência foi a de 60%, todavia, gerou indivíduos de baixa qualidade entre 850 e 840. Com a taxa a 70% os melhores indivíduos foram gerados, variando entre 830 e 820. A taxa definida a 90% demorou a convergir, ultrapassando as 7 mil gerações e gerou indivíduos de baixa qualidade.

**Quadro 4.3** - Soluções encontradas para o problema Caixeiro Viajante

Soluções para o problema Caixeiro Viajante						
Taxas de Cruzamento	Algoritmo genético de Holland		Algoritmo genético com o operador proposto - Um ponto de Corte		Algoritmo genético com o operador proposto - Dois pontos de Corte	
	Qualidade dos indivíduos	Gerações	Qualidade dos indivíduos	Gerações	Qualidade dos indivíduos	Gerações
60%	865	368	839	778	846	2089
70%	842	237	805	779	825	3781
90%	829	485	808	385	857	7277

**Fonte:** (Elaborado pelo Autor, 2022)

No quadro 4.3 é possível observar de forma resumida os resultados expostos nos gráficos 4.4, 4.5 e 4.6. Sendo possível notar que, para a taxa de cruzamento fixada em 60% e em 70% o resultado que gerou os melhores indivíduos foi o do algoritmo genético com o operador proposto com um ponto de corte, entretanto, o resultado que obteve melhor convergência foi o algoritmo de Holland. De tal forma, o mais recomendado entre os três AG seria o AG com operador proposto com um ponto de corte, visto que há um impasse de não existir uma solução que melhor satisfaça a condição de possuir a melhor qualidade de indivíduos e o menor número de gerações. Então, partindo da definição do problema de

querer o melhor caminho em que o custo seja minimal, esse valor pode ser encontrado a partir do AG de um ponto de corte.

Quando a taxa de cruzamento foi fixada em 90%, o melhor resultado tanto de qualidade de indivíduos, quanto de convergência, foi obtido por meio do AG com o operador proposto com um ponto de corte. Enquanto que o algoritmo genético com dois pontos de cortes apresentou indivíduos não satisfatórios, como também mais de 7 mil gerações para realizar a convergência no AG.

Nota-se que o algoritmo de dois pontos de corte não obteve resultado satisfatório, mesmo mantendo os blocos de construção que existiam nos pais em seus filhos. Essa dificuldade se dá por conta de que, com o passar das gerações todos os cromossomos da população tendem a pertencer ao mesmo *schema*, logo fica difícil encontrar dois pontos de cortes para realizar o cruzamento.

Nota-se que o algoritmo que obteve melhor desempenho para o problema do caixeiro viajante, foi o algoritmo com o operador proposto (3.2) com um ponto de corte, com taxa de cruzamento de 90%. Pois, por meio dessa combinação, o AG conseguiu obter resultados satisfatórios na qualidade de indivíduos, como também, consegue realizar a convergência antes de 400 gerações.

No próximo capítulo serão encontradas as considerações finais sobre esse trabalho, como também os possíveis trabalhos futuros que podem ser realizados por meio do operador de cruzamento desenvolvido neste trabalho.

## CAPÍTULO 5

### CONCLUSÃO

Apesar da grande quantidade de estudo que existe na área dos algoritmos genéticos, muitos autores propõem a criação de um novo operador de cruzamento de tal forma que analisam apenas os *schemas* que estão presentes na população. De modo que, a preservação dos blocos de construções presentes entre os pais ainda não é devidamente explorada.

Tal exploração, ainda pode não ser realizada pelo fato do algoritmo genético de Holland com o operador de um ponto de corte, ainda conseguir demonstrar resultados satisfatórios para os problemas de otimização funcional e combinatória, conseguindo a geração de bons indivíduos candidatos à solução e convergência satisfatória.

Entretanto, ao realizar a implementação do operador de cruzamento proposto nota-se que o algoritmo genético consegue gerar indivíduos com qualidade superior ao algoritmo de Holland, como também, auxilia o algoritmo genético a convergir mais rápido.

Como foi demonstrado por Fan et al. (2005) e Zhang et al. (2007) que novos operadores de cruzamento realizando a conservação dos blocos de construção existentes na população, auxiliou a manter os genes mais importantes da população preservados. A conservação desses genes tornam-se essenciais para otimizar o desempenho do AG.

Contudo, ao manter os blocos de construção e aplicar o cruzamento de dois pontos, é notório o mal desempenho do AG. Pois, em um dado momento, todos os indivíduos irão pertencem ao mesmo *schema* e será difícil encontrar mais de um ponto de corte para o realizar o cruzamento.

Logo, este trabalho conseguiu demonstrar por meio de testes empíricos, que a conservação dos blocos de construção existentes entre os pais para o cruzamento de um ponto

de corte, auxiliou na qualidade dos resultados e no tempo de convergência do AG. Corroborou também que o cruzamento com dois pontos de cortes, onde a conservação dos blocos de construção é realizada, não demonstrou resultados satisfatórios nos problemas de otimização a que foi submetido.

## **5.1 TRABALHOS FUTUROS**

Recomenda-se para trabalhos futuros a aplicação deste operador sobre mais problemas de otimização, para validar as conclusões obtidas neste trabalho.

Como também, realizar a análise da taxa de mutação, tendo em vista que a mesma foi fixada em 10%. Tornando-se interessante a análise de um possível hibridismo entre o operador proposto nesta pesquisa com um algoritmo genético auto-adaptativo. A fim de, analisar a influência da mutação sobre o operador proposto nesta pesquisa, tanto por meio da qualidade dos indivíduos, como da convergência do algoritmo.

## REFERÊNCIAS BIBLIOGRÁFICAS

AIZAWA, A. N.; WAH, B. W. *Scheduling of Genetic Algorithms in a Noisy Environment*. *Evolutionary Computation*, 2(2), 97–122, 1994. doi:10.1162/evco.1994.2.2.97.

BARCELLOS, J.C.H. **Algoritmos Genéticos Adaptativos: Um estudo comparativo**. 2000. 143 p. Dissertação (Mestre em Engenharia) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2000.

CARVALHO, C.P.L.F. **Algoritmos Genéticos**. O Instituto de Ciências Matemáticas E de Computação (ICMC) Da USP, sites.icmc.usp.br/andre/research/genetic/#intro.

CARVALHO, L. F. B. S. **Um Jogo de Damas Evolutivo**. 95 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Federal de Alagoas, Maceió, 2008.

CATARINA, A. S.; BACH, S. L. **Estudo do efeito dos parâmetros genéticos sobre a solução otimizada e sobre o tempo de convergência em algoritmos genéticos com codificações binária e real**. *Acta Scientiarum*, v. 25, n. 2, p. 147–152, 2003.

CHEBBI, O. ; CHAOUACHI, J. *Effective parameter tuning for genetic algorithm to solve a real world transportation problem*. *20th International Conference on Methods and Models in Automation and Robotics (MMAR)*. doi:10.1109/mmar.2015.7283904, 2015

DARWIN, C. **A Origem das Espécies**, Trad. do original: *On the Origin of Species*. Hemus Editora LTDA - SP, 1959.

EIBEN, A E.; SMITH, J. E. *Introduction to Evolutionary Computing*. Berlin, Heidelberg Springer Berlin Heidelberg, 2015.

ENAMI, L. M., et al. **O Problema do caixeiro viajante através de algoritmo genético**, 2019.

FAN, L., et al. *A new crossover operator based on the rough set theory for genetic algorithms*. *2005 International Conference on Machine Learning and Cybernetics*, 2005, pp. 2907-2912 Vol. 5, doi: 10.1109/ICMLC.2005.1527439.

GALVAN, B., et al. *Parallel Evolutionary Computation for Solving Complex CFD Optimization Problems : A Review and Some Nozzle Applications*. *Parallel Computational Fluid Dynamics* 2002, no. 9780444506801, 2003, pp. 573–604, [www.sciencedirect.com/science/article/pii/B9780444506801500723](http://www.sciencedirect.com/science/article/pii/B9780444506801500723), 10.1016/b978-044450680-1/50072-3. Accessed 10 May 2022.

GIGUERE, P.; GOLDBERG, D. E. *Population sizing for optimum sampling with genetic algorithms: A case study of the Onemax problem*. *Genetic Programming*, v. 98, p. 496-503, 1998

GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989

GOLDBERG, D. E. *The Design of Innovation. Genetic Algorithms and Evolutionary Computation*. 2002. doi:10.1007/978-1-4757-3643-4

GOOGLE. *Google Collaboratory*. [S. l.], 2017. Disponível em: <https://colab.research.google.com>.

GUEDES, A. C. B.; LEITE, J. N. F.; ALOISE, D. J. **Um algoritmo genético com infecção viral para o problema do caixeiro viajante**. *Revista Publica*, v. 1, n. 1, 2005.

GUIMARÃES, T. S. M. **Apoio à Síntese De Modelos Estruturais De Software Orientado A Objetos Utilizando Algoritmos Genéticos Co-Evolucionários**. 2005. 127 p. Dissertação (Mestrado em Engenharia Elétrica) - PUC-Rio, Rio de Janeiro, 2005

HOLLAND, J.H. *Adaptation in natural and artificial systems*. MIT Press, 1975

HOLLAND, J.H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

HOLLAND, J.H. *Building Blocks, Cohort Genetic Algorithms, and Hyperplane-Defined Functions*. *Evolutionary Computation*, 8(4), 373–391, 2000. doi:10.1162/106365600568220

KATO, R., et al. **Algoritmos Genéticos**. In: *BIOINFO - Revista Brasileira de Bioinformática e Biologia Computacional*. 1ª ed.; vol. 1. Lagoa Santa: Editora Alfahelix, 2021. DOI: 10.51780/978-6-599-275326

KAZADI, S. T. *Conjugate Schema and Basis Representation of Crossover and Mutation Operators*. *Evolutionary Computation*, 1998, 6(2), 129–160. doi:10.1162/evco.1998.6.2.129

KOZA, J.R. *Genetic programming as a means for programming computers by natural selection*. *Stat Comput* 4, 87–112 (1994). <https://doi.org/10.1007/BF00175355>

LINDEN, Ricardo. **Algoritmos genéticos (2a edição)**. Brasport, 2008.

Mendel, G. *Versuche über Pflanzen-hybriden*. *Verhandlungen des naturforschenden Ver-eines in Brünn*, Bd. IV für das Jahr 1865, Abhand-lungen, 3–47 (1866).

MITCHELL, M. *An Introduction to Genetic Algorithms*. Cambridge, Mass. ; London, Mit, 1998.

NUNES, J. R. O. **Avaliação de Taxas de Cruzamento E Mutação Em Um Algoritmo Genético Baseado Em Ordem Aplicado Ao Problema Do Caixeiro Viajante**. Orientador: Flavius da Luz e Gorgônio. 2018. 60 p. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Universidade Federal do Rio Grande do Norte – UFRN, Caicó, 2018.

RODRIGUES, H.; MACHADO, N. C. S. **Otimização De Funções Polinomiais Utilizando Conceitos De Cálculo Diferencial E Algoritmos Genéticos**. 9 Dec. 2019, p. 64,

ROSA, T. O.; LUZ, H. S. **Conceitos básicos de algoritmos genéticos: teoria e prática**. Anais do XI encontro de estudantes de informática do Tocantins, p. 27–37, 2009.

SOARES, A.H.M. **Algoritmos de estimação de distribuição baseados em árvores filogenéticas**. 196 p. Tese (Doutorado em Ciências da Computação e Matemática Computacional) - USP - São Carlos, São Carlos, 2014.

VILHENA, R.V.L. **Um algoritmo genético baseado em tipos abstratos de dados e sua especificação em Z**. 2003. 209 p. Tese Doutorado (Doutorado em Ciência da Computação) - UFPE, Recife, 2003.

VON ZUBEN, Fernando J. **Computação evolutiva: uma abordagem pragmática**. Tutorial: Notas de Aula da disciplina IA707, Faculdade de Engenharia Elétrica e de Computação-Universidade Estadual de Campinas, 2000.

WELDON, W. F. R. *Mendel's Laws of Alternative Inheritance in Peas*. *Biometrika* 1, no. 2 (1902): 228–54. <https://doi.org/10.2307/2331488>.

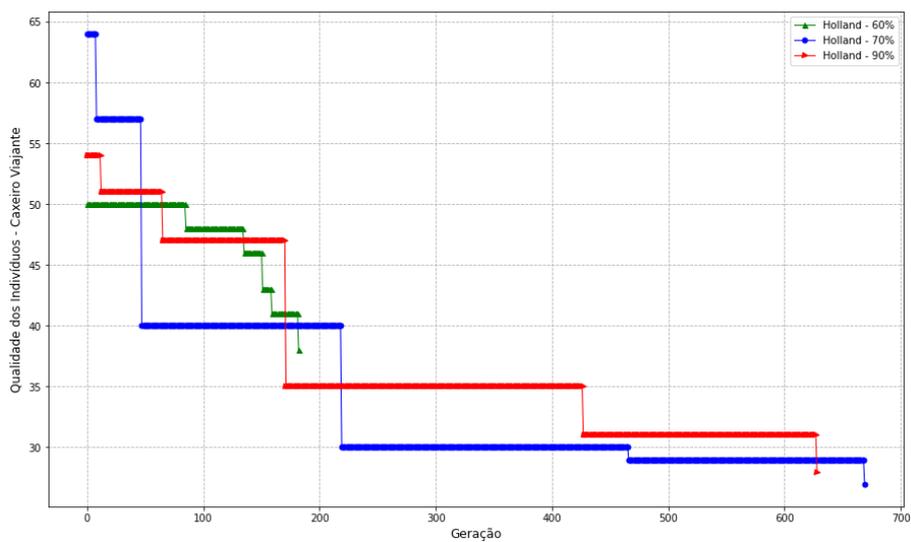
YUAN, S., et al. *A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms*. *European Journal of Operational Research*, 2013, 228(1), 72–82. doi:10.1016/j.ejor.2013.01.043

ZHANG, M., et al. *A new crossover operator in genetic programming for object classification*. *IEEE Trans Syst Man Cybern B Cybern*. 2007; 37(5):1332-43. doi: 10.1109/tsmcb.2007.902043. PMID: 17926713

## APÊNDICE

### Apêndice A - Resultado AG de Holland para o Caixeiro Viajante

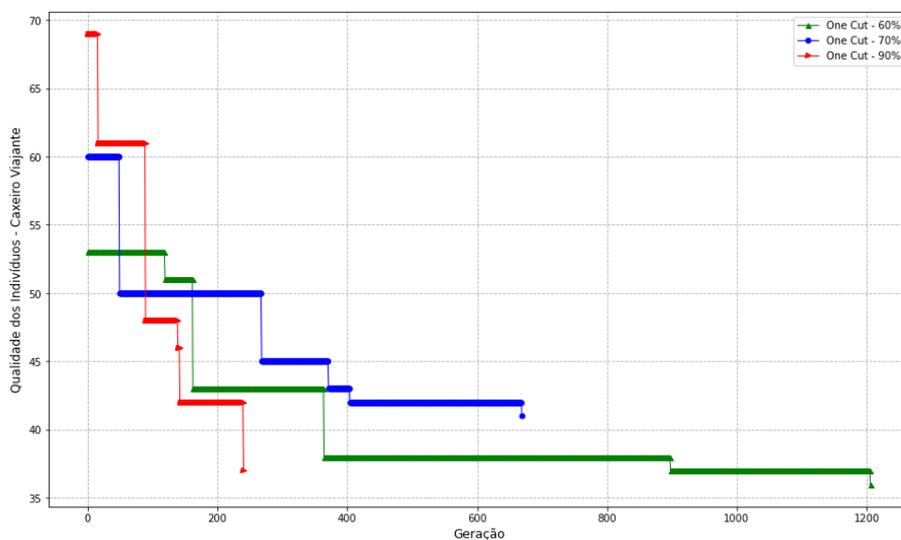
- Tamanho da população = 30
- Tamanho do cromossomo = 10



Fonte: (Elaborado pelo Autor, 2022)

## Apêndice B - Resultado cruzamento proposto com um ponto de corte para o Caixaero Viajante

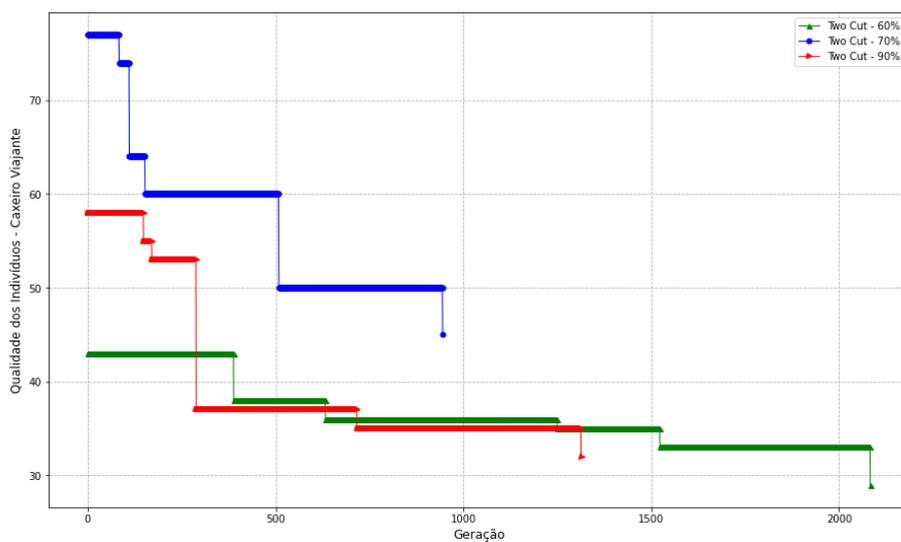
- Tamanho da população = 30
- Tamanho do cromossomo = 10



Fonte: (Elaborado pelo Autor, 2022)

## Apêndice C - Resultado cruzamento proposto com dois pontos de cortes para o Caixaero Viajante

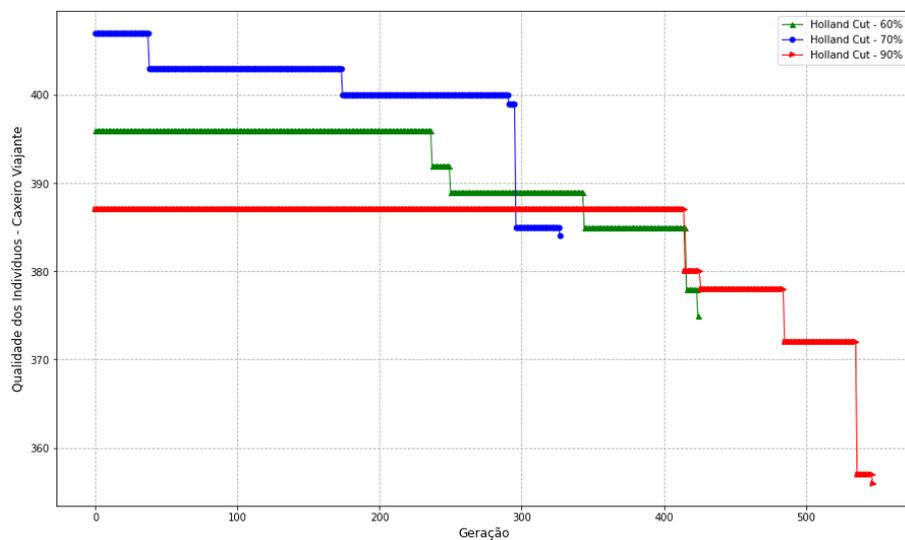
- Tamanho da população = 30
- Tamanho do cromossomo = 10



Fonte: (Elaborado pelo Autor, 2022)

### Apêndice D - Resultado AG de Holland para o Caixeiro Viajante

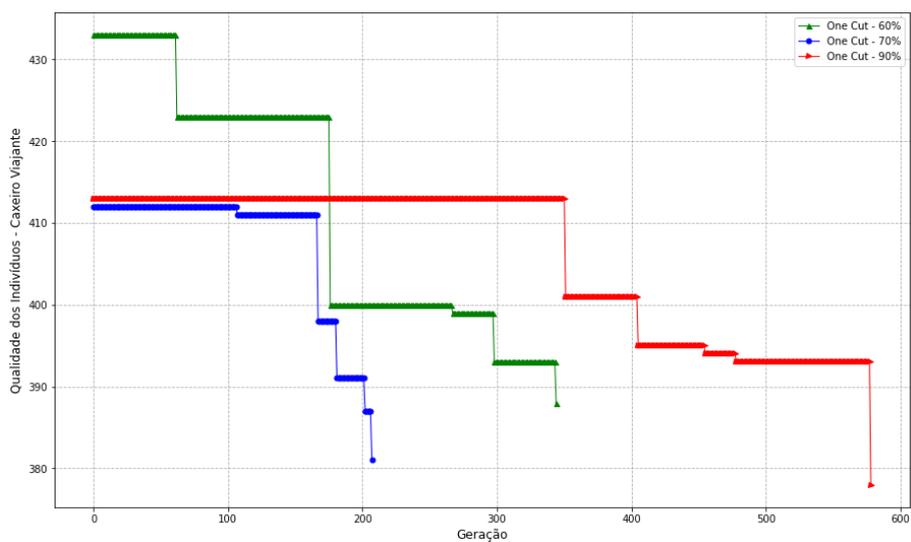
- Tamanho da população = 30
- Tamanho do cromossomo = 50



Fonte: (Elaborado pelo Autor, 2022)

### Apêndice E - Resultado cruzamento proposto com um ponto de corte para o Caixeiro Viajante

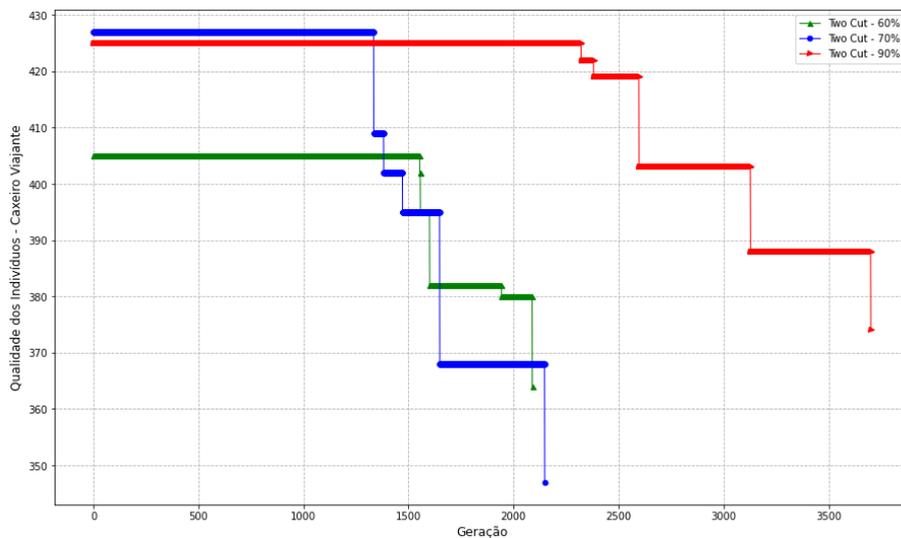
- Tamanho da população = 30
- Tamanho do cromossomo = 50



Fonte: (Elaborado pelo Autor, 2022)

## Apêndice F - Resultado cruzamento proposto com dois pontos de cortes para o Caixeiro Viajante

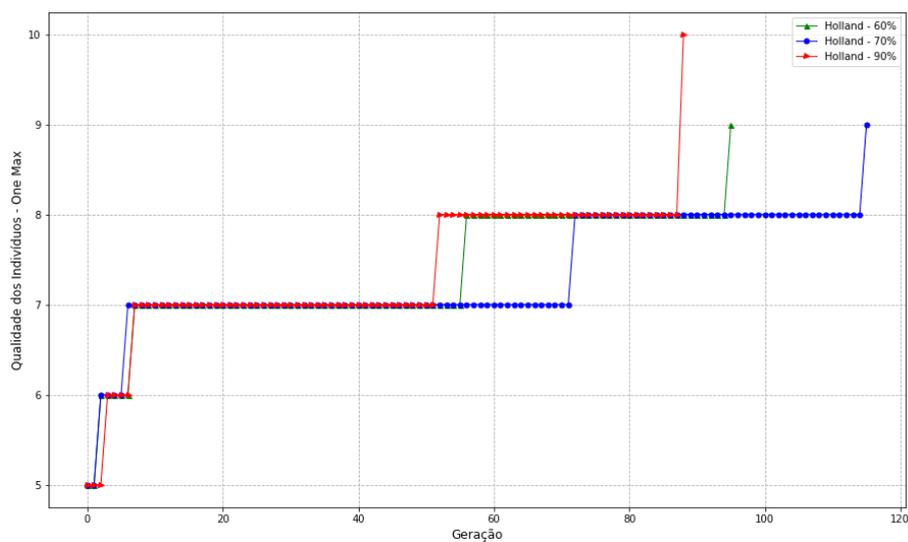
- Tamanho da população = 30
- Tamanho do cromossomo = 50



Fonte: (Elaborado pelo Autor, 2022)

## Apêndice G - Resultado AG de Holland para OneMax

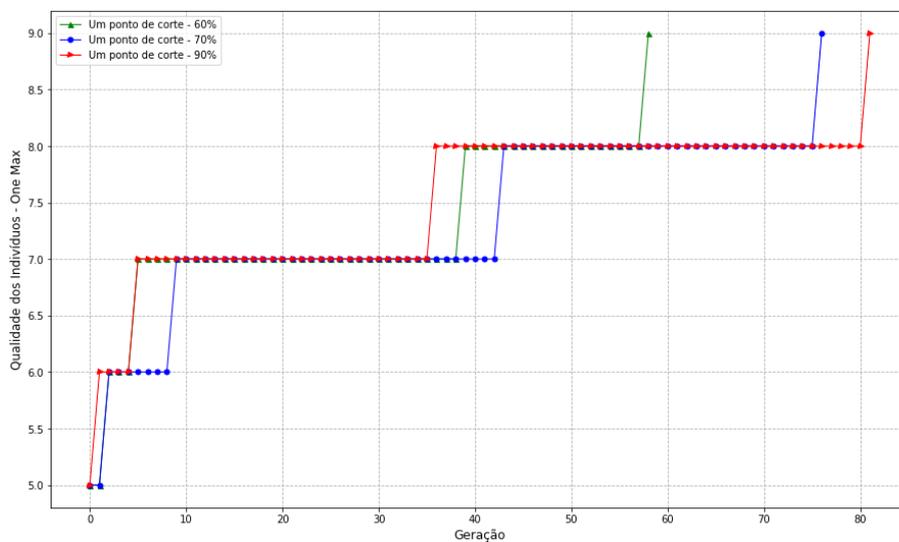
- Tamanho da população = 30
- Tamanho do cromossomo = 10



Fonte: (Elaborado pelo Autor, 2022)

## Apêndice H - Resultado cruzamento proposto com um ponto de corte para OneMax

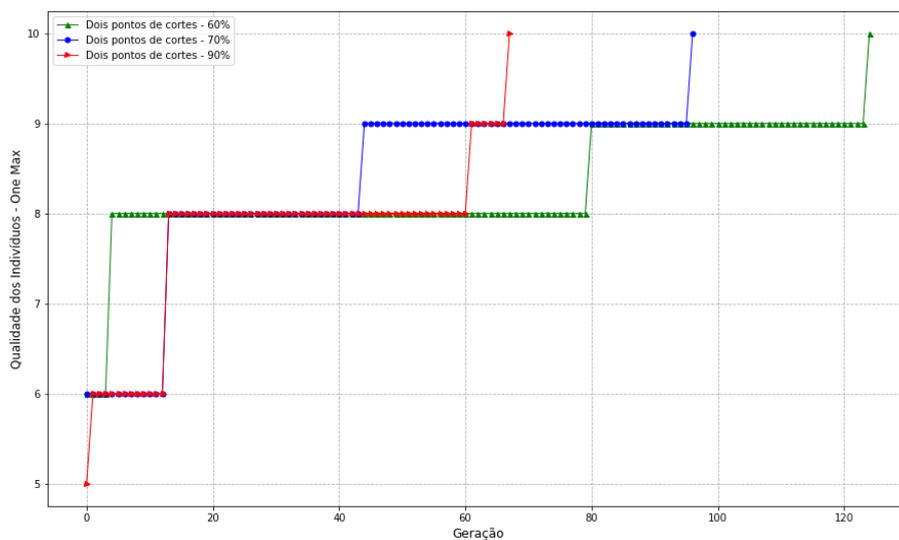
- Tamanho da população = 30
- Tamanho do cromossomo = 10



Fonte: (Elaborado pelo Autor, 2022)

## Apêndice I - Resultado cruzamento proposto com dois pontos de cortes para OneMax

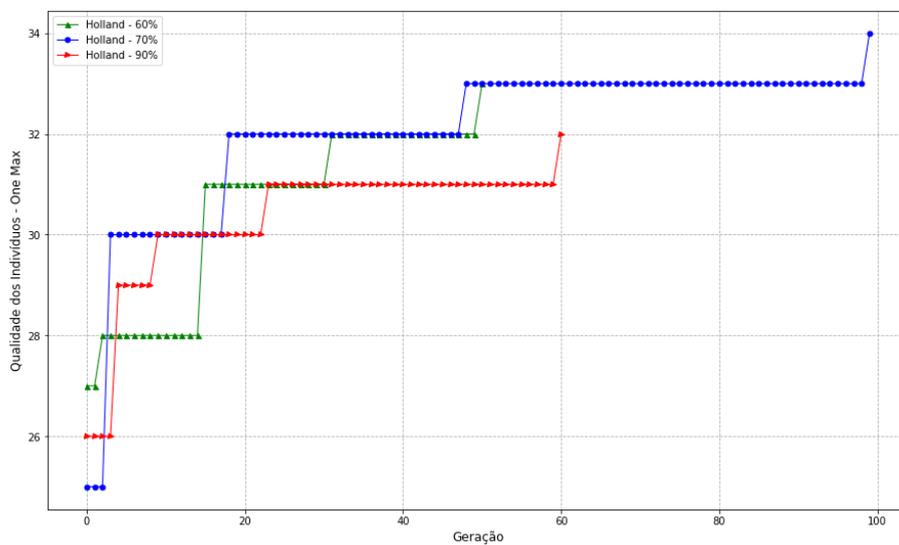
- Tamanho da população = 30
- Tamanho do cromossomo = 10



Fonte: (Elaborado pelo Autor, 2022)

### Apêndice J - Resultado AG de Holland para OneMax

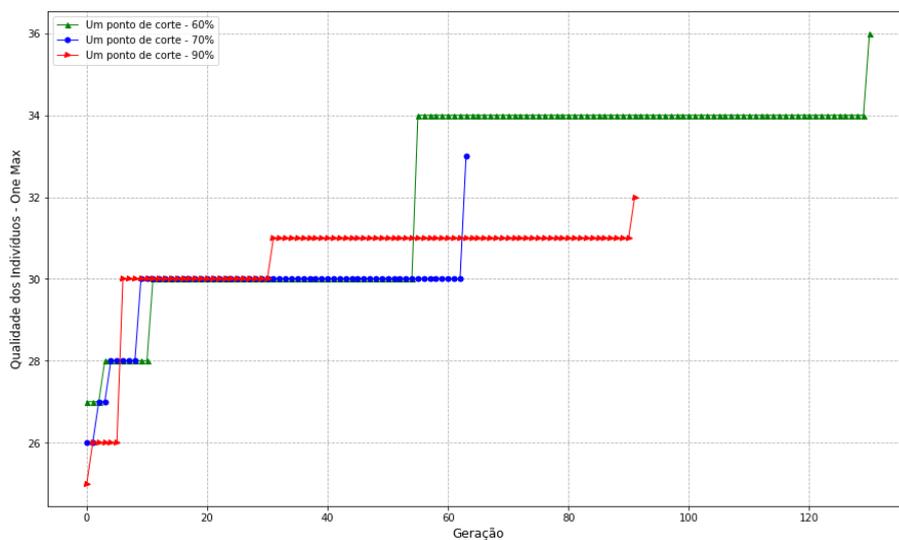
- Tamanho da população = 30
- Tamanho do cromossomo = 50



Fonte: (Elaborado pelo Autor, 2022)

### Apêndice K - Resultado cruzamento proposto com um ponto de corte para OneMax

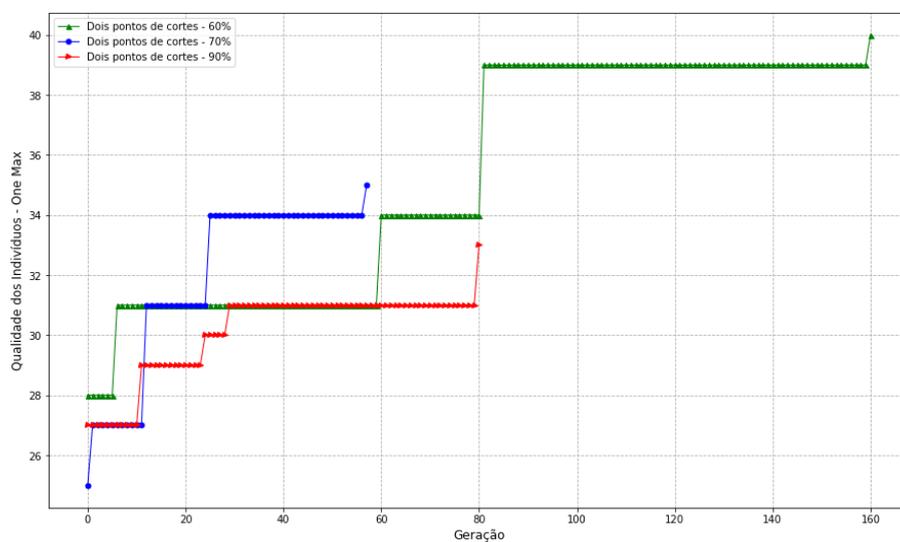
- Tamanho da população = 30
- Tamanho do cromossomo = 50



Fonte: (Elaborado pelo Autor, 2022)

## Apêndice L- Resultado cruzamento proposto com dois pontos de cortes para OneMax

- Tamanho da população = 30
- Tamanho do cromossomo = 50



**Fonte:** (Elaborado pelo Autor, 2022)