

Mike Lucas Souza Paiva

**Investigação do uso da regra de um quinto de  
sucesso no operador de *crossover* do AG.**

Timóteo

2017

Mike Lucas Souza Paiva

**Investigação do uso da regra de um quinto de sucesso no  
operador de *crossover* do AG.**

Monografia apresentada à Coordenação de Engenharia de Computação do Campus Timóteo do Centro Federal de Educação Tecnológica de Minas Gerais para obtenção do grau de Bacharel em Engenharia de Computação.

Centro Federal de Educação Tecnológica de Minas Gerais

Campus Timóteo

Graduação em Engenharia de Computação

Orientador: Douglas Nunes de Oliveira

Timóteo

2017

Mike Lucas Souza Paiva

## **Investigação do uso da regra de um quinto de sucesso no operador de *crossover* do AG.**

Monografia apresentada à Coordenação de Engenharia de Computação do Campus Timóteo do Centro Federal de Educação Tecnológica de Minas Gerais para obtenção do grau de Bacharel em Engenharia de Computação.

Trabalho aprovado. Timóteo, 28 de março de 2017:



---

**Douglas Nunes de Oliveira**  
Orientador



---

**Prof. Marcelo de Sousa Balbino**  
Professor Convidado



---

**Prof. João Batista Queiroz Zuliani**  
Professor Convidado

Timóteo  
2017

*Dedico esse trabalho aos meus pais,  
Reinaldo (in memoriam) e Edilene, fonte de motivação e educação.*

# Agradecimentos

Agradeço primeiramente a Deus, o qual permitiu que tudo isso acontecesse, e não somente nesta etapa, mas todas as coisas em minha vida. Por me dar a graça de viver, a força e saúde para superar as minhas limitações.

Aos meus pais, Reinaldo (*in memoriam*) e Edilene, por todo amor, proteção, exemplo de vida e motivação dado a mim. Aos meus irmãos, Marcus e Magno, pelos momentos de companhia.

Ao meu orientador e professor Douglas, pela confiança deste trabalho, conhecimento adquirido e prontidão de me ajudar sempre.

A toda minha família, que sempre me incentivou manter o foco com fé, força e coragem. Em especial, Vó Nazinha e Tia Nilce, pela perseverança e orações. A Fernanda, minha prima querida, pelos conselhos e momentos de lazer.

Ao Fábio, companheiro de diversas horas, solidariedade inefável e por sempre apoiar as minhas decisões.

A todos os meus amigos, que entenderam e estiveram ao meu lado durante esse período de "correria" e sempre me ajudaram nos estudos e trabalhos: Ariella, Bruna, Jackson, Jordana, Karini, Paula, Tamires, Tássio e Thiago.

Aos professores e funcionários do CEFET-MG campus Timóteo, que me proporcionaram, não somente, o conhecimento racional, mas também manifestação de caráter afetividade de educação no processo de formação profissional.

A todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

# Resumo

O algoritmo genético (AG) é considerado uma das técnicas de busca mais utilizada em diversas áreas de estudos, principalmente na computação. Essas técnicas visam encontrar soluções para problemas de otimização. John Henry Holland foi o precursor nos estudos de AG. Inspirado na teoria da evolução, o algoritmo genético pode representar o processo natural de adaptação dos indivíduos dentro de um ambiente. As técnicas evolutivas permitem que seus parâmetros se adaptem durante as gerações. A proposta desta pesquisa é investigar o uso de uma técnica evolutiva, denominada como regra um quinto de sucesso, durante o processo de recombinação do AG. Tal regra foi desenvolvida por Rechenberg em 1973, a qual parte da ideia que um quinto dos filhos gerados devem ser melhores que seus pais. E nesta condição, foi adaptado a implementação do algoritmo genético para alcançar o comportamento do parâmetro evolutivo. A adaptação foi no operador genético de *crossover*, variando a taxa de recombinação dos genes nos indivíduos. Foi estudada então duas implementações, uma elaborada pelo orientador deste trabalho, Douglas Nunes de Oliveira e estudada no trabalho da Paula Peçanha Gonçalves, denominada como FGA (*full genetic algorithm*); e uma outra com a regra de um quinto de sucesso no FGA, intitulada de FGA+RS. Para avaliar os indivíduos, são utilizadas funções de avaliações de otimização que visa minimizar os resultados. Ambas implementações foram executadas para seis funções, uma de cada vez. Para cada função, foi criada uma população inicial, preenchida com valores aleatórios no intervalo de cada função. O resultado da execução, é uma população final com os melhores indivíduos. Os valores finais, foram comparados e analisados, pelos critérios de menor e maior valor, valor médio e pelo teste de Wilcoxon pareado, que se trata de um método estatístico. Embora, ambas implementações operem com os mesmos processos de seleção, mutação e recombinação, a adaptação feita, em comparação com a original, apresentou resultados consideráveis para a função Dixon-Price, observados a partir de teste estatístico.

**Palavras-chave:** algoritmo genético, operador de *crossover*, regra de um quinto de sucesso, técnica evolutiva.

# Abstract

A genetic algorithm (GA) it is known to be one of the search techniques most used in different study fields, especially when we adventure in the computational field. These techniques are focused in finding solutions to optimization problems. One of the representations to the GA, a field of studies led by John Henry Holland and inspired by the theory of evolution, it is the representation as the natural process of individuals adaptation to an environment. The evolutionary techniques allows its parameters to adapt along with the generations. This research proposes a investigation regarding the use of a evolutionary technique, known as 1/5th rule, while running the recombination process of the GA. The 1/5th rule was developed by Rechenberg in 1973, and it is based on the concept that one fifth of the generated children should be more fit than the parents. At this state, the implementation of the GA was adapted to reach the evolutionary parameter behavior. The adaptation was applied to the genetic operator crossover, varying the rate of genes in the individuals. Then, a study was made on two implementations: the first one was elaborated by this research's advisor, Douglas Nunes de Oliveira, and studied by Paula Peçanha Gonçalves, defined as FGA (full genetic algorithm); the second one it is an implementation based on the 1/5th rule applied to the FGA, named FGA+RS. In the evaluation of individuals, optimization evaluation function are used in order to minimize the results. Both implementations were applied on six functions, one by one. To each function, the initial population was populated by random values within the interval of each function. The result is a final population with the most fit individuals. The final results are compared and analyzed by lower and higher metrics, the average value and the Wilcoxon signed rank test, an statistical hypothesis test. Although both implementations operate with the same selection process, mutation and recombination, the adaptation, if compared with the original one, has shown relevant results to the Dixon-Price function, observed from the statistical test.

**Keywords:** genetic algorithm, recombination operator, 1/5th rule.

# Lista de ilustrações

Figura 1 – Esquema do ciclo evolutivo de um algoritmo genético. . . . .	15
Figura 2 – Exemplo de roleta viciada em uma população de cinco indivíduos. . . .	18
Figura 3 – Exemplo de <i>crossover</i> de um ponto. . . . .	18
Figura 4 – Exemplo de <i>crossover</i> de dois pontos. . . . .	19
Figura 5 – Exemplo de <i>crossover</i> uniforme. . . . .	19
Figura 6 – Exemplo de <i>crossover</i> aritmético. . . . .	20
Figura 7 – Gráfico da função Griewank . . . . .	23
Figura 8 – Gráfico da função Rastrigin . . . . .	23
Figura 9 – Gráfico da função Schwefel . . . . .	24
Figura 10 – Gráfico da função Dixon Price . . . . .	25
Figura 11 – Gráfico da função Rosenbrock . . . . .	25
Figura 12 – Gráfico da função Levy . . . . .	26
Figura 13 – Representação gráfica do <i>boxplot</i> . . . . .	29
Figura 14 – Representação gráfica do <i>boxplot</i> para o exemplo . . . . .	30
Figura 15 – Fluxograma do FGA . . . . .	35
Figura 16 – Fluxograma do FGA+RS . . . . .	37
Figura 17 – <i>Boxplots</i> do <i>R-project</i> para o primeiro teste . . . . .	42
Figura 18 – <i>Boxplots</i> do <i>R-project</i> para o segundo teste . . . . .	43
Figura 19 – Gráfico dos tempos de execução . . . . .	45

# Lista de tabelas

Tabela 1 – Exemplo de amostras de teste Wilcoxon pareado . . . . .	27
Tabela 2 – Diferenças ordenadas pelo seu valor absoluto. . . . .	28
Tabela 3 – Resumo do primeiro teste . . . . .	39
Tabela 4 – Resultados do primeiro experimento no teste Wilcoxon pareado . . . . .	40
Tabela 5 – Mediana para primeiro teste . . . . .	41
Tabela 6 – Resumo do segundo teste . . . . .	42
Tabela 7 – Resultados do segundo experimento no teste Wilcoxon pareado . . . . .	43
Tabela 8 – Tempo de execução dos testes . . . . .	44
Tabela 9 – Tempo de execução para tamanho de população diferentes . . . . .	44
Tabela 10 – Dados amostrais do melhor indivíduo aplicado na função Griewank do primeiro teste . . . . .	51
Tabela 11 – Dados amostrais do melhor indivíduo aplicado na função Rastrigin do primeiro teste . . . . .	52
Tabela 12 – Dados amostrais do melhor indivíduo aplicado na função Schwefel do primeiro teste . . . . .	53
Tabela 13 – Dados amostrais do melhor indivíduo aplicado na função Dixon-Price do primeiro teste . . . . .	54
Tabela 14 – Dados amostrais do melhor indivíduo aplicado na função Rosenbrock do primeiro teste . . . . .	55
Tabela 15 – Dados amostrais do melhor indivíduo aplicado na função Levy do primeiro teste . . . . .	56
Tabela 16 – Dados amostrais do melhor indivíduo aplicado na função Griewank do segundo teste . . . . .	57
Tabela 17 – Dados amostrais do melhor indivíduo aplicado na função Rastrigin do segundo teste . . . . .	58
Tabela 18 – Dados amostrais do melhor indivíduo aplicado na função Schwefel do segundo teste . . . . .	59
Tabela 19 – Dados amostrais do melhor indivíduo aplicado na função Dixon-Price do segundo teste . . . . .	60
Tabela 20 – Dados amostrais do melhor indivíduo aplicado na função Rosenbrock do segundo teste . . . . .	61
Tabela 21 – Dados amostrais do melhor indivíduo aplicado na função Levy do segundo teste . . . . .	62

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>1.1</b>	<b>Justificativa</b>	<b>12</b>
<b>1.2</b>	<b>Objetivo</b>	<b>12</b>
<b>1.3</b>	<b>Organização do trabalho</b>	<b>13</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>14</b>
<b>2.1</b>	<b>Algoritmos Genéticos</b>	<b>14</b>
2.1.1	Representação de indivíduo	15
2.1.2	População inicial	16
2.1.3	Função de Avaliação	16
2.1.4	Seleção de pais	17
2.1.5	Operadores genéticos	17
2.1.5.1	Operador de <i>Crossover</i>	18
2.1.5.2	Mutação	20
2.1.5.3	Seleção da nova população	21
<b>2.2</b>	<b>Estratégia evolucionária (EE)</b>	<b>21</b>
2.2.0.1	Regra de 1/5 de sucesso	21
<b>2.3</b>	<b>Funções <i>benchmark</i></b>	<b>22</b>
2.3.0.1	Função Griewank	22
2.3.0.2	Função Rastrigin	22
2.3.0.3	Função Schwefel	24
2.3.0.4	Função Dixon Price	24
2.3.0.5	Função Rosenbrock	24
2.3.0.6	Função Levy	25
<b>2.4</b>	<b>Teste de Wilcoxon pareado</b>	<b>26</b>
<b>3</b>	<b>PROCEDIMENTOS METODOLÓGICOS</b>	<b>31</b>
<b>4</b>	<b>IMPLEMENTAÇÃO</b>	<b>32</b>
<b>4.1</b>	<b>Implementação dos AGs</b>	<b>32</b>
4.1.1	Implementação FGA	33
4.1.2	Implementação FGA+RS	34
<b>5</b>	<b>RESULTADOS</b>	<b>39</b>
<b>5.1</b>	<b>Apresentação dos resultados</b>	<b>39</b>
5.1.1	Primeiro teste	39

5.1.2	Segundo teste . . . . .	41
5.1.3	Tempo de execução . . . . .	41
<b>6</b>	<b>CONCLUSÃO . . . . .</b>	<b>46</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>47</b>
	<b>APÊNDICES</b>	<b>50</b>
	<b>APÊNDICE A – TABELAS DO PRIMEIRO TESTE . . . . .</b>	<b>51</b>
	<b>APÊNDICE B – TABELAS DO SEGUNDO TESTE . . . . .</b>	<b>57</b>

# 1 Introdução

Algoritmos genéticos (AG) são utilizados na otimização de problema em várias áreas do conhecimento. Como por exemplo, diversos trabalhos científicos utilizam AG para solucionar os problemas propostos, tais como: [Ferreira \(2015\)](#) utilizou AG para solucionar o problema de colisões e alocação de horários institucionais para o CEFET-MG, capaz de solucionar e eliminar os conflitos dos horários entre professores, alunos, disciplinas e a disponibilidade de recursos (salas, laboratórios); [Rocha \(2015\)](#) empregou o AG no problema de formação de células de manufatura, conhecido como Manufatura Celular, que consiste em formar blocos diagonais em uma matriz de incidência, representando quais peças passam por quais máquinas em seu processo de manufatura; [Goldbarg, Goldbarg e Neto \(2005\)](#) propuseram uma configuração de custo mínimo para um sistema de cogeração de energia que utiliza o gás natural como fonte energética, utilizando algoritmos evolucionários desenvolvidos de acordo com as metáforas dos algoritmos genéticos, algoritmos meméticos e da transgenética computacional. Trabalhos com o parâmetros de auto-ajuste também são relevantes na computação: [Silva \(2010\)](#) propôs uma estratégia de autoadaptação de parâmetros para o algoritmo de Evolução Diferencial. Esta estratégia deve fazer uso de parâmetros adequados e utilizar operadores para a mutação dos parâmetros que levam em consideração o "conhecimento" adquirido, durante o processo de busca; [Arruda e Kagan \(2006\)](#) estudaram a utilização de algoritmos da área de Sistemas Inteligentes, explicitamente Algoritmos Evolutivos (AEs) baseados em Estratégias Evolutivas (EEs), para estimação do estado harmônico de uma rede elétrica. Além destes trabalhos, o "*IEEE Congress on Evolutionary Computation*" (IEEE CEC) é um dos principais eventos na área de computação evolutiva, e abrange todos os tópicos em computação evolutiva da teoria às aplicações ([TAN; YEN, 2016](#)).

O estudo sobre AG foi formalmente elaborado por John Henry Holland ([HOLLAND, 1975](#)), inspirado na Teoria da Evolução das Espécies, propondo um modelo heurístico computacional para otimizar e melhorar indivíduos ao longo de gerações. Para [Holland \(1975\)](#) uma solução candidata é um indivíduo na biologia. Ele define ainda que partes da solução candidata é um cromossomo e que partes do cromossomo é um gene. Para simulação do processo natural de evolução, os algoritmos evolucionários (AE) utilizam modelos computacionais para resolver problemas. Apesar de existir uma diversidade de variantes de AE, todos os modelos possuem uma similaridade na execução da evolução das espécies através de seleção, mutação e reprodução, tais processos estão diretamente ligados ao comportamento dos indivíduos no meio onde estão presentes.

Os processos de seleção, mutação e recombinação existentes na biologia, são utilizados como operadores genéticos nos AGs. O conjunto de soluções candidatas encontradas

aplicando os operadores genéticos é chamado de geração. Na computação, cada gene do indivíduo pode ser representado por valores binário, inteiros ou reais. Esses operadores possuem grande relevância na convergência dos indivíduos. Ao longo de gerações, há um indivíduo que represente uma solução satisfatória ou ótima. Sendo um consenso entre muitos pesquisadores, o AG, no início do seu processo, deve investigar de forma mais ampla o espaço de busca e no final do processo refine as soluções candidatas ou indivíduos produzidos ao longo do processo evolucionário.

No desenvolvimento das estratégias evolucionárias (EE), [Rechenberg \(1973\)](#) define que um indivíduo pode representar uma solução candidata de um problema utilizando valores reais, além dos seus operadores possuírem funcionalidades interessantes aos AGs com representação real ([LINDEN, 2012](#)). Quando se utiliza esse tipo de representação, podemos aplicar a técnica conhecida como "um quinto de sucesso" no operador de mutação e dessa forma produzir dinamicamente o efeito de busca amplificada e busca localizada, conforme o comportamento dos indivíduos durante o processo evolutivo.

Esta técnica consiste em, de tempos a tempos, ao longo do processo de busca, calcular a melhora dos filhos em relação aos pais, isto é, a razão entre o número de sucessos e o número de iterações. Se a razão for maior que  $1/5$ , aumenta a variância, se for menor que  $1/5$ , diminua-se a variância. Tal estratégia é denominada como evolucionária, pois auxilia na velocidade de convergência ao valor ótimo. Além disto possui a característica de aumentar ou restringir a busca no espaço de soluções.

## 1.1 Justificativa

O desafio recorrente por diversos estudiosos e que este trabalho também enfrenta, é possibilitar um AG mais suscetível à solução de um problema ou que possa trabalhar com uma performance melhor. Com este intuito, uma abordagem é adaptar os operadores genéticos, ajustando as estratégias evolutivas aplicadas em operadores de *crossover*. Dessa forma, com a motivação de se pesquisar melhorias no AG em sua diversidade de aplicação, foi proposto uma adaptação na regra de  $1/5$  de sucesso no processo de recombinação no AG. Tal modificação permite que a quantidade de genes recombinados seja alterada dinamicamente, tentando causar o mesmo efeito proporcionado no operador de mutação. Técnica semelhante a esta não foi encontrada nos operadores de recombinação para representações reais.

## 1.2 Objetivo

Partindo do princípio apontado por [Rechenberg \(1973\)](#), onde a regra de um quinto de sucesso, quando aplicada no operador de mutação, auxilia na convergência do algoritmo,

foi levantada a hipótese de uma variante desta regra beneficiar o operador de *crossover*, tomando como parâmetro a quantidade de genes.

Neste sentido, tem-se como objetivo investigar a adaptação da regra de um quinto de sucesso aplicada no processo de recombinação dos AGs, com o desejo de resultar em uma convergência satisfatória do que já se tem presente no FGA (GONÇALVES, 2016) A fim de atingir este objetivo, será aplicada uma variante da regra de um quinto de sucesso no operador de *crossover* aritmético.

### 1.3 Organização do trabalho

Este trabalho apresenta uma investigação do uso da regra de um quinto de sucesso aplicada no operador *crossover* do AG. Posteriormente um comparativo dos resultados de uma implementação já elaborada em outro trabalho com a adaptação do uso da regra de 1/5 como variante. O mesmo está estruturado em seis capítulos.

No primeiro capítulo trata do levantamento do problema a ser estudado neste trabalho, detalhando sua motivação, relevância e seu objetivo.

No segundo capítulo está a história e levantamentos bibliográficos que fundamentam os assuntos relevantes para a compreensão deste trabalho. Nele são apresentados os algoritmos genéticos, um breve histórico do mesmo, definição, seus operadores genéticos; um estudo sobre estratégia evolutiva; as funções *benchmark* utilizadas para avaliação dos AGs; além de apanhado sobre o teste de Wilcoxon pareado.

No capítulo 3 está presente as etapas que envolveram desde o estudo e compreensão de AG até o desenvolvimento do trabalho.

No quarto capítulo aborda a implementação do trabalho. Nessa seção é discorrido sobre o algoritmo elaborado, as diferenças entre as implementações e a adaptação da proposta.

Os resultados obtidos estão apresentados no capítulo 5. Um resumo dos teste realizados. Para cada função e implementação utilizada é tabelado e analisado pelo teste de Wilcoxon pareado.

O capítulo 6 finaliza este trabalho com a conclusão da proposta e algumas ideias de projetos futuros a serem desenvolvidos a partir deste.

Os dados amostrais por completo estão apresentados seção de apêndices. Para cada teste feito, os dados foram organizados em tabelas com os melhores indivíduos encontrados pelas implementações, separadas por função de avaliação, uma para cada.

## 2 Fundamentação teórica

Nesta seção serão abordados os fundamentos teóricos para a compreensão dos métodos utilizados neste trabalho, onde serão apresentados aspectos gerais sobre algoritmos genéticos, operadores genéticos, a regra um quinto de sucesso e as funções *benchmark* aplicadas no AG.

### 2.1 Algoritmos Genéticos

O AG começou a ser estudado por [Holland \(1975\)](#) e hoje possui diversas áreas de aplicações na engenharia, matemática, bioinformática ([NALDI, 2006](#)) e principalmente em pesquisas computacionais. Foi posto por ele que os algoritmos genéticos são heurísticas de busca baseados no processo de evolução natural de uma população.

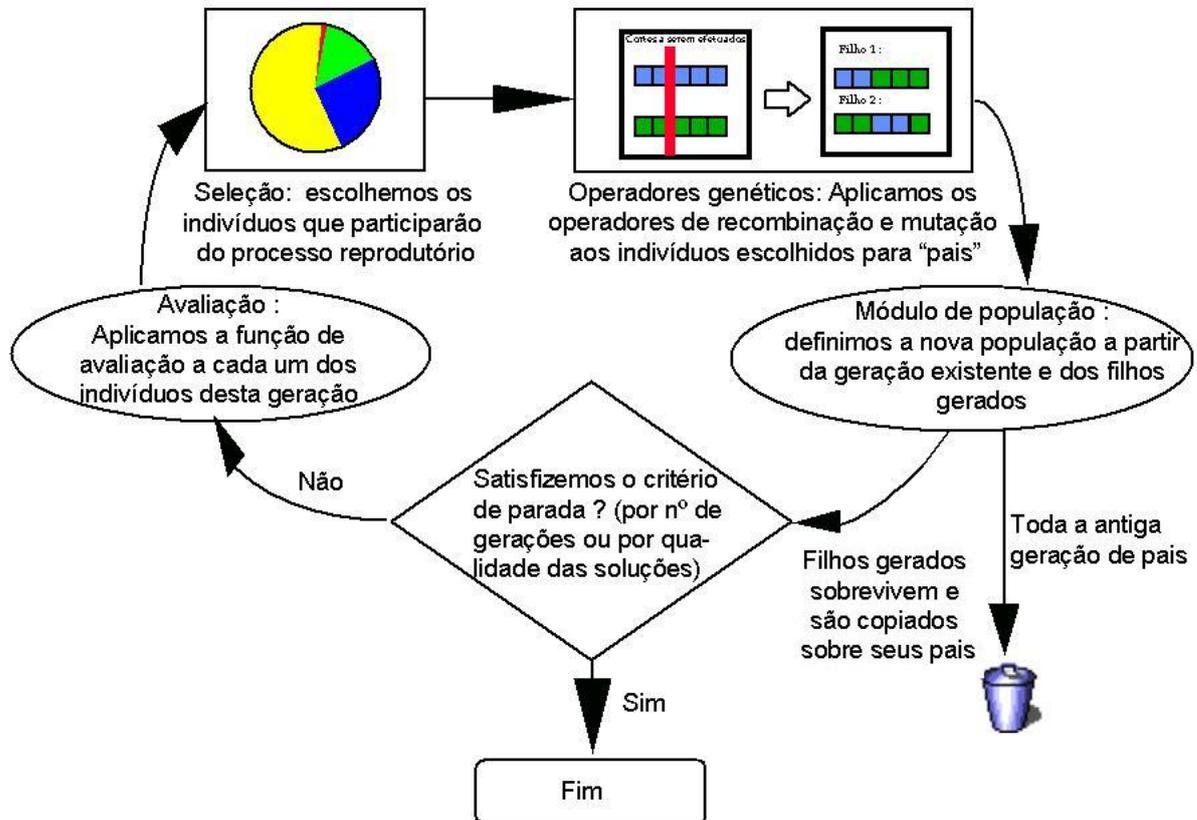
O autor [Linden \(2012\)](#) apresenta de forma simples e prática os algoritmos genéticos e seus principais elementos. Posto por ele: "Algoritmos genéticos (AG) são um ramo dos algoritmos evolucionários e como tal podem ser definidos como uma técnica de busca baseada na metáfora do processo biológico de evolução natural." ([LINDEN, 2012](#)). Tal relação de semelhança consiste pela representação de que uma população de soluções candidatas, chamadas de indivíduos, são submetidas ao processo de busca ou processo evolutivo. Os indivíduos são criados e sujeitos aos processos dos operadores genéticos: seleção, recombinação e mutação, para que as características qualitativas dos indivíduos possam ser melhoradas a cada geração.

Esse ciclo do algoritmo genético pode ser representado pela Figura 1. Além do esquema do ciclo evolutivo, [Linden \(2012\)](#) resumiu o algoritmo genético nos principais passos:

1. Inicialização da população de indivíduos;
2. Avaliação de cada indivíduo na população;
3. Seleção dos pais para gerar novos indivíduos;
4. Aplicação dos operadores de crossover e mutação destes pais de forma a gerar os indivíduos da nova geração;
5. Exclusão dos velhos membros da população;
6. Avaliação de todos os novos indivíduos e inserção na população;

7. Se o tempo acabou, ou o melhor indivíduo satisfaz os requerimentos e desempenho, retorne-o. Caso contrário, volte para o passo 3).

Figura 1 – Esquema do ciclo evolutivo de um algoritmo genético.



Fonte: (LINDEN, 2012)

### 2.1.1 Representação de indivíduo

Um indivíduo da população é a representação de uma solução candidata para o problema em questão. Conforme na biologia, tem-se o gene, que se trata de cada parte do indivíduo, onde estão contidas suas características. Segundo Linden (2012), para obter um bom funcionamento e resultado do algoritmo genético é necessário que o modelo do indivíduo passado para o computador seja bem definido e adequado ao problema real. Desta forma, Linden (2012), destacou algumas regras gerais:

1. Simplificação da representação;
2. Se houver soluções proibidas, não é necessário a representação das mesmas;
3. Representação de condições do problema, caso exista.

Inicialmente os indivíduos eram representados por valores binários (HOLLAND, 1975), neste caso, o indivíduo é composto por uma cadeia binária, onde cada gene assumirá valores de 0 ou 1. Tal representação é amplamente utilizada por pesquisadores da área de algoritmos genéticos. A representação com números reais também é possível nos AG, para isso as estratégias evolucionárias foram estudadas (RECHENBERG, 1973).

### 2.1.2 População inicial

Uma população é representada por um conjunto de indivíduos, a mesma passará pelo processo de evolução natural. Entretanto, para gerar uma população inicial é importante pensar em uma técnica em que os indivíduos estejam uniformemente distribuídos por todo espaço de busca, garantindo que tenha uma variedade genética.

Linden (2012) sugere que existe uma grande chance de cobrir uniformemente todo o espaço de busca, ao se escolher aleatoriamente os indivíduos da população inicial. Segundo ele, isso acontece devido a lei das probabilidades.

### 2.1.3 Função de Avaliação

A função de avaliação, é considerada a forma de avaliar o indivíduo qualitativamente. (HOLLAND, 1975) definiu a performance de um indivíduo como *fitness*, ou seja, valor indicativo de quão apto está o indivíduo. Além disso, a função de avaliação, tem o efeito de escolher as melhores soluções para o problema.

O fato da função de avaliação avaliar a qualidade de um indivíduo, sem levar em consideração as técnicas usadas para resolver o problema, Linden (2012) define a mesma como a única relação do algoritmo com o problema real. Por esse motivo, o AG tem a capacidade de calcular o máximo de qualquer função de  $n$  variáveis, sem ter a necessidade de alterar o algoritmo, apenas a própria função de avaliação.

Linden (2012) também cita alguns pontos relevantes sobre o *fitness*:

- Para elaborar a função de avaliação, deve-se conhecer o problema real, suas restrições e os objetivos de qualidade;
- Se um indivíduo  $a_1$  apresentar uma solução melhor que um indivíduo  $a_2$ , então a avaliação do  $a_1$  deve ser maior do que o  $a_2$ ;
- Os resultados encontrados pela função de avaliação devem diferenciar entre as soluções subótimas, permitindo classificar qual está mais próxima da solução desejada.

### 2.1.4 Seleção de pais

A seleção dos pais trata-se do processo de eleger os indivíduos que serão selecionados para a próxima geração, este método deve simular o mecanismo de seleção natural nos seres da biologia.

O objetivo é privilegiar os pais com melhores *fitness*, sem ter que desprezar os pais que possuem uma avaliação menor. Porém, a seleção de pais não deve ser baseada somente na escolha dos melhores indivíduos, pois existe a possibilidade de um indivíduo com uma aptidão menor possuir propriedades genéticas favoráveis à geração de um indivíduo que possua a melhor solução para o problema em questão, considerando que tais propriedades favoráveis não estejam presentes nos demais indivíduos da população (ROSA; LUZ, 2009).

Além disso, de acordo com Linden (2012), se apenas os melhores indivíduos forem selecionados, a população gerada tenderá a ser compostas por indivíduos cada vez mais semelhantes, não havendo uma diversidade dos mesmos, e não permitindo uma evolução satisfatória. Os principais métodos de seleção são o da "roleta viciada" (*roulette wheel*), seleção por torneio e o elitismo.

No método da roleta viciada cada indivíduo da população é representado na roleta proporcionalmente à sua aptidão. Assim, quanto maior for o *fitness* do indivíduo, maior será sua "fatia" da roleta, enquanto os indivíduos que apresentarem um *fitness* menor é dada uma "fatia" relativamente menor da roleta. Com os indivíduos distribuídos na roleta Figura 2, ela é girada um determinado número de vezes, e são escolhidos os indivíduos sorteados na roleta.

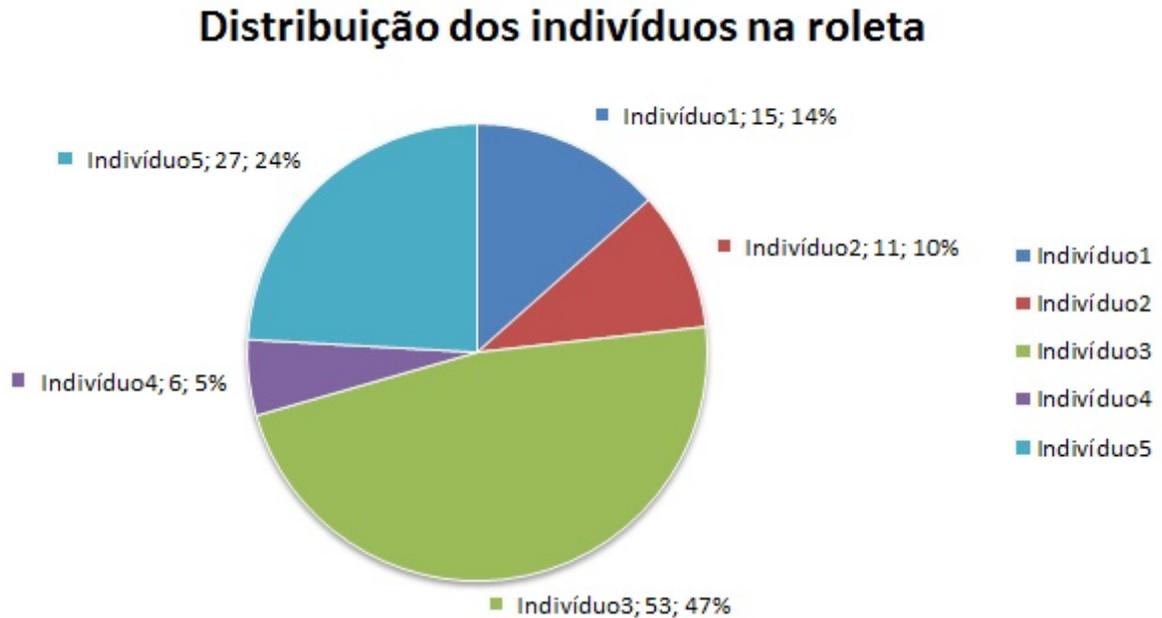
No método de seleção por torneio, é feito uma competição do fitness de cada indivíduo com os demais indivíduos, escolhidos aleatoriamente. O que possuir melhor *fitness*, ganha a disputa. Após todos disputarem, os ganhadores das disputas são selecionados para a aplicação do operador genético (LINDEN, 2012).

O método de elitismo é a seleção dos melhores indivíduos dentre as populações. Por este método, o desempenho do AG pode ser aumentado, pois previne a perda das melhores soluções encontradas (OBITKO, 1998).

### 2.1.5 Operadores genéticos

Os operadores genéticos tem como função aumentar a diversidade dos indivíduos da população, gerar indivíduos com capacidade melhor, contribuindo assim para que as populações evoluam a cada nova geração (ROSA; LUZ, 2009).

Figura 2 – Exemplo de roleta viciada em uma população de cinco indivíduos.



Fonte: o autor

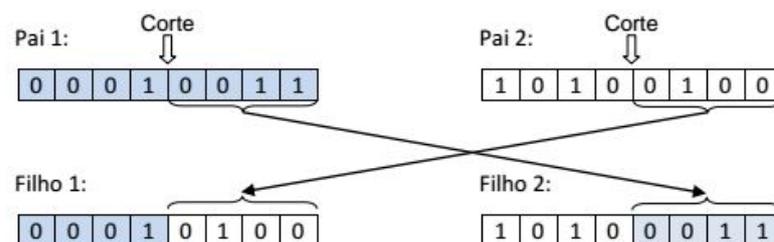
#### 2.1.5.1 Operador de *Crossover*

O operador de *crossover*, também conhecido como de recombinação, é o procedimento responsável pela troca de características dos pais durante o processo de reprodução, gerando filhos e permitindo que as próximas gerações herdem essas características. A ideia do *crossover* é garantir que os indivíduos descendentes sejam melhores que seus pais, se caso eles herdarem as melhores características de cada pai (CAZANGI, 2002).

Há diversos métodos de *crossover*, os mais conhecidos são o *crossover* de um ponto, *crossover* de dois ou mais pontos, o *crossover* uniforme e o *crossover* aritmético.

O *crossover* de um ponto é o corte mais simples dos existentes, nele é definido um ponto e feito a troca entre os dois pais e gerando assim os filhos, Figura 3.

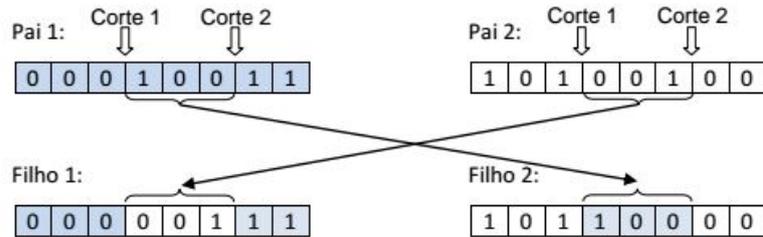
Figura 3 – Exemplo de *crossover* de um ponto.



Fonte: o autor

No *crossover* de dois pontos são feitos dois cortes, definindo um intervalo entre o ponto inicial (primeiro corte) e o ponto final (segundo corte). Este intervalo é trocado entre os pais, gerando os filhos, Figura 4.

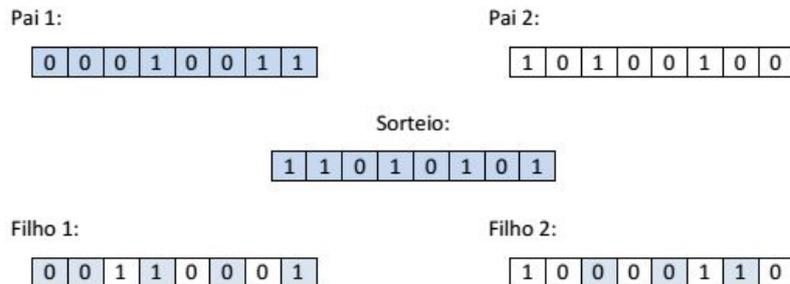
Figura 4 – Exemplo de *crossover* de dois pontos.



Fonte: o autor

No *crossover* uniforme é feito um sorteio, entre zero e um, para cada gene. Se o valor sorteado for igual a um, o primeiro filho recebe o gene, na mesma posição, do primeiro pai, e o segundo filho recebe o gene do segundo pai. Caso contrário, se o valor sorteado foi igual a zero, o primeiro filho recebe o gene, na mesma posição, do segundo pai, e o segundo filho recebe o gene do primeiro pai, Figura 5.

Figura 5 – Exemplo de *crossover* uniforme.

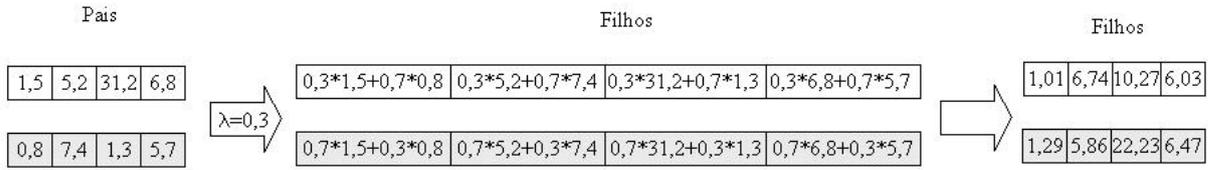


Fonte: o autor

No *crossover* aritmético é definido previamente um valor  $\lambda$ , sendo que  $\lambda \in [0, 1]$ , desta forma são gerados descendentes a partir de uma combinação linear dos indivíduos da população pai ( $P_1$  e  $P_2$ ). A combinação linear que gera os indivíduos filhos  $F_1$  e  $F_2$  é dada pelas equações 2.1 e 2.2, respectivamente, (GONÇALVES, 2013). Um exemplo pode ser ilustrado na Figura 6. Para este tipo de *crossover* utiliza a representação real para os genes dos indivíduos.

$$F_1 = \lambda P_1 + (1 - \lambda) P_2 \tag{2.1}$$

$$F_2 = \lambda P_2 + (1 - \lambda) P_1 \tag{2.2}$$

Figura 6 – Exemplo de *crossover* aritmético.

Fonte: (LINDEN, 2012)

### 2.1.5.2 Mutação

O operador de mutação é importante para o AG, uma vez que ele permite a diversidade entre as populações. Além disso, o processo de mutação permite que o AG faça uma busca mais ampla sobre as populações e na possibilidade de encontrar soluções fora dos limites da população inicial.

Linden (2012), realça como deve ser escolhido a intensidade da mutação, caso seja pequena demais (poucos genes), por não gerar uma diversidade nos indivíduos a convergência não surgiria tanto efeito. Da mesma forma, se a intensidade de alteração fosse alta (muitos genes), a população irá perder características de indivíduos importantes.

As principais formas de mutação existente são: a mutação pontual, a mutação uniforme e a mutação gaussiana.

A mutação pontual é utilizado quando se aplica a representação binária nos indivíduos. Nesta mutação, são sorteados alguns genes e estes têm seu valor alterado (GOLDBERG; HOLLAND, 1988).

A mutação uniforme (MICHALEWICZ; SCHOENAUER, 1996) é aplicada em indivíduos de representação real. Seu processo é iniciado a partir da escolha aleatória de um gene e aplicado um novo valor para este gene selecionado. O valor aleatório selecionado deve estar dentro do intervalo permitido (COELHO et al., 2000).

Assim como a mutação uniforme, a mutação gaussiana, é utilizada para indivíduos representados por valores reais. Além disto, nas estratégias evolutivas este operador é baseado em uma distribuição de probabilidades normal ou gaussiana de média zero e desvio padrão ( $\sigma$ ) (LINDEN, 2012). A fórmula da distribuição normal é dada por Linden (2012, p. 280):

$$N(0, \sigma, u) = \frac{e^{-\frac{1}{2}(\frac{u}{\sigma})^2}}{\sigma\sqrt{2\pi}} \quad (2.3)$$

onde  $u$  é um número real positivo entre zero e um.

O gene selecionado,  $x_i$ , é somado a seu valor o resultado da distribuição normal.

$$x'_i = x_i + N(0, \sigma, x) \quad (2.4)$$

### 2.1.5.3 Seleção da nova população

Para que o AG continue procurando uma solução em novos indivíduos, é feita a seleção da nova população, ou seja, são definidos os indivíduos que passarão para a próxima geração.

A técnica do elitismo leva em consideração o valor do *fitness* da população atual e os novos indivíduos gerados pelos operadores genéticos. A ideia básica do elitismo é a seguinte: "os  $k$  melhores indivíduos de cada geração não devem 'morrer' junto com a sua geração, mas sim passar para a próxima visando garantir que seus genomas sejam preservados." (LINDEN, 2012).

## 2.2 Estratégia evolucionária (EE)

Nos anos 60, foi proposto por Rechenberg (1965) e Schwefel (1975), estratégias evolucionárias que permitam solucionar problemas de otimização baseadas em parâmetros Rechenberg (1973). Apesar das estratégias terem sido estudadas em paralelo com os AGs, elas possuem a representação real que usa um operador de mutação baseado em uma distribuição normal (LINDEN, 2012).

### 2.2.0.1 Regra de 1/5 de sucesso

A regra de um quinto de sucesso, foi criada por Rechenberg (1973), ele propõe atualizar o(s) parâmetro(s) de estratégia ao longo das gerações, pois caso seja um valor constante, as estratégias evolucionária eventualmente irá convergir para uma solução ótima, porém não há limite de tempo que isso possa acontecer, posto por Linden (2012).

Um exemplo de parâmetro que pode ser utilizado no operador de mutação é o desvio padrão ( $\sigma$ ). A ideia principal desta regra parte do princípio que um quinto dos filhos gerados devem ser melhores do que seus pais. Se o índice de melhoria for menor do que um quinto, pode se concluir que a solução está próxima do máximo local e a busca deve ser feita em passos menores, ou seja, o parâmetro de evolução desvio padrão ( $\sigma$ ) deve diminuir.

Linden (2012) define que, caso o índice de melhora for maior do que um quinto, pode se dizer que a solução está distante do máximo e a busca deve ser feita em passos maiores, ou seja, o parâmetro de evolução de desvio padrão ( $\sigma$ ) deve aumentar. O efeito da aplicação desta técnica pode acelerar a convergência do algoritmo (EIBEN; SMITH, 2003). A regra pode ser resumida na Equação 2.5.

$$\sigma = \begin{cases} \sigma, p_s = 1/5 \\ \sigma \cdot c, p_s < 1/5 \\ \sigma/c, p_s > 1/5 \end{cases} \quad (2.5)$$

Onde:

- $p_s$  é o índice de melhora nas últimas  $n$  gerações;
- $c$  é um parâmetro do algoritmo definido de forma *ad hoc*, normalmente é um valor de  $0,817 \leq c \leq 1$  (SCHWEFEL, 1981).

## 2.3 Funções *benchmark*

As funções *benchmark* são utilizadas para testes de algoritmos de otimização. Elas são agrupadas de acordo com as semelhanças em suas propriedades físicas significativas e formas (SURJANOVIC; BINGHAM, 2015). As funções relevantes para este trabalho são as de  $d$  dimensões e por apelo comparativo, as mesmas utilizadas na proposta de Gonçalves (2016). As seguintes funções foram utilizadas neste trabalho.

### 2.3.0.1 Função Griewank

A função Griewank possui vários mínimos locais amplamente espalhados e que são distribuídos de forma regular. A complexidade dessa função é mostrada nos gráficos com zoom na Figura 7. Sua expressão é apresentada na Equação 2.6, onde  $d$  é o tamanho da dimensão,  $x_i \in [-600, 600]$ , para todo  $i = 1, \dots, d$  e seu mínimo global é  $f(x^*) = 0$ , em  $x^* = (0, \dots, 0)$ .

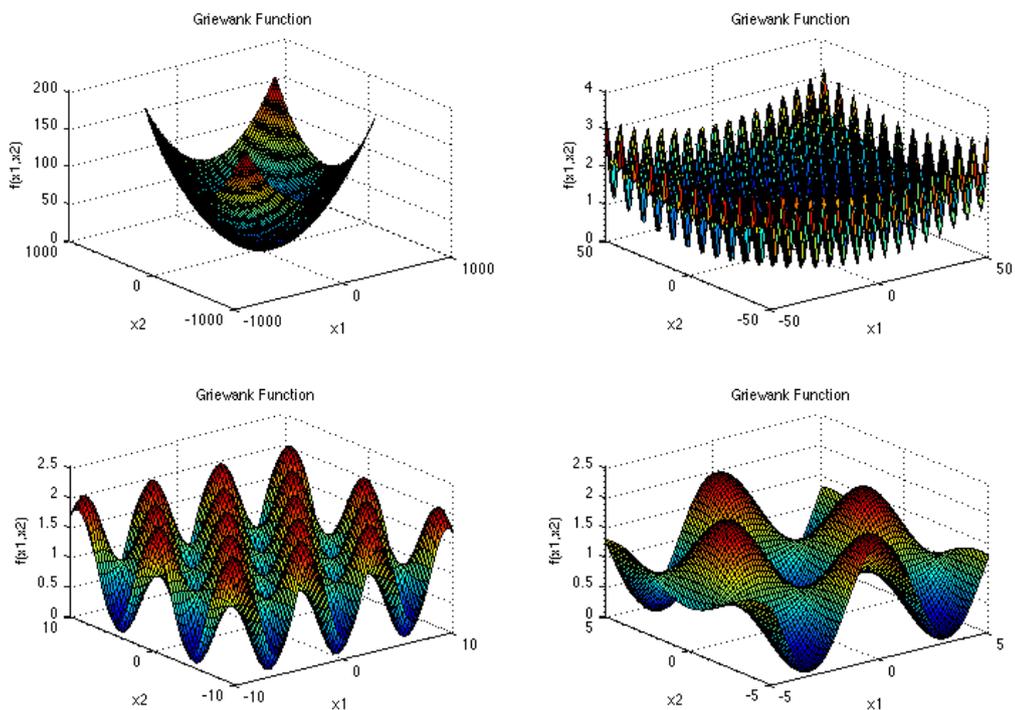
$$f(x) = \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (2.6)$$

### 2.3.0.2 Função Rastrigin

A função Rastrigin possui vários mínimos locais. É altamente multimodal, mas as localizações dos mínimos locais são regularmente distribuídos, podendo ser visualizado na Figura 8. Sua expressão é apresentada na Equação 2.7, onde  $d$  é o tamanho da dimensão,  $x_i \in [-5.12, 5.12]$ , para todo  $i = 1, \dots, d$  e seu mínimo global é  $f(x^*) = 0$ , em  $x^* = (0, \dots, 0)$ .

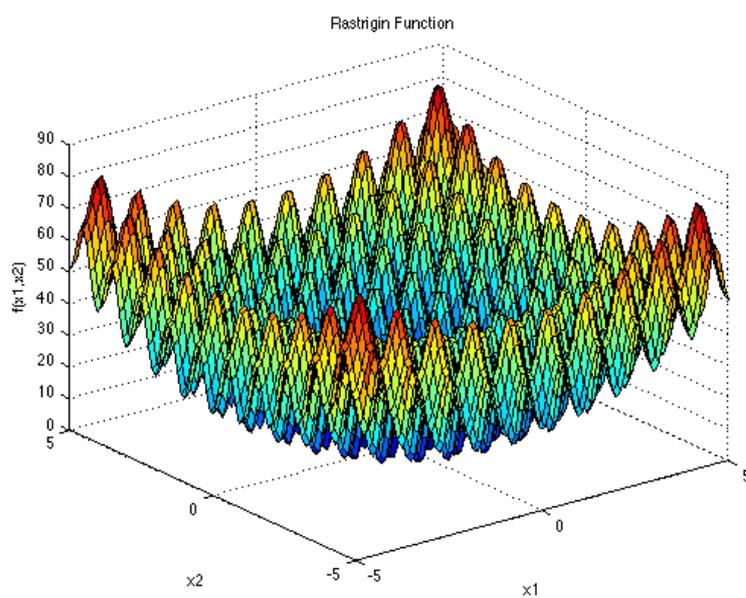
$$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)] \quad (2.7)$$

Figura 7 – Gráfico da função Griewank



Fonte: (SURJANOVIC; BINGHAM, 2015)

Figura 8 – Gráfico da função Rastrigin



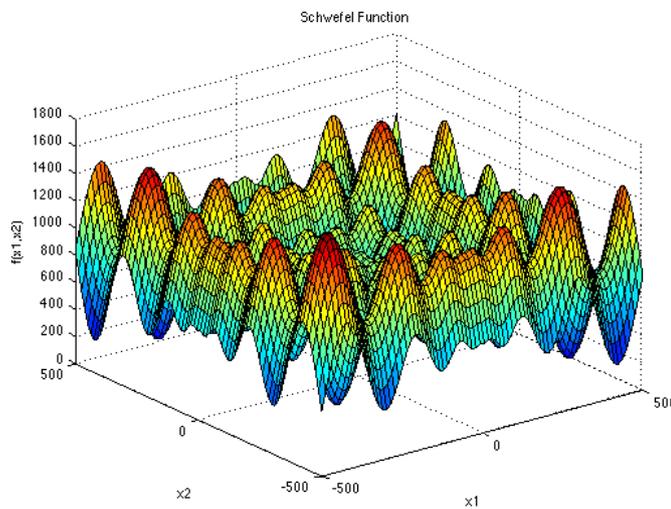
Fonte: (SURJANOVIC; BINGHAM, 2015)

### 2.3.0.3 Função Schwefel

A função Schwefel é complexa, com vários mínimos locais. O gráfico anterior, Figura 9, mostra a forma bidimensional da função. Sua expressão é apresentada na Equação 2.8, onde  $d$  é o tamanho da dimensão,  $x_i \in [-500, 500]$ , para todo  $i = 1, \dots, d$  e seu mínimo global é  $f(x^*) = 0$ , em  $x^* = (420.9687, \dots, 420.9687)$ .

$$f(x) = 418.9829d - \sum_{i=1}^d x_i \sin(\sqrt{|x_i|}) \quad (2.8)$$

Figura 9 – Gráfico da função Schwefel



Fonte: (SURJANOVIC; BINGHAM, 2015)

### 2.3.0.4 Função Dixon Price

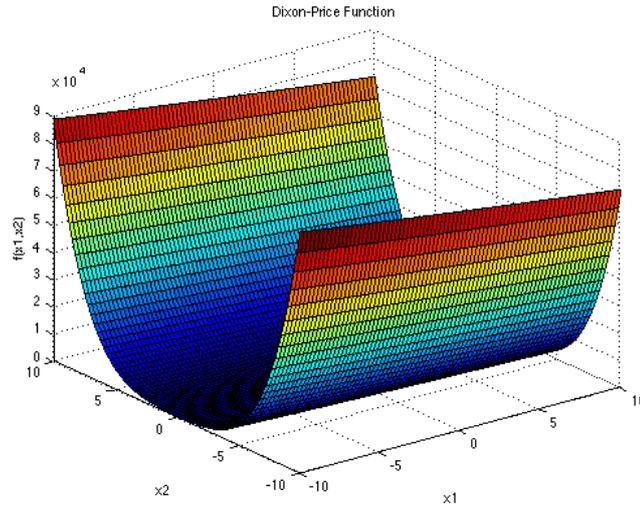
A função Dixon Price é ilustrada na Figura 10. Sua expressão é apresentada na Equação 2.9, onde  $d$  é o tamanho da dimensão,  $x_i \in [-10, 10]$ , para todo  $i = 1, \dots, d$  e seu mínimo global é  $f(x^*) = 0$  em  $x_i = 2^{\frac{-2^i - 2}{2^i}}$  para todo  $i = 1, \dots, d$ .

$$f(x) = (x_1 - 1)^2 + \sum_{i=2}^d i(2x_i^2 - x_{i-1})^2 \quad (2.9)$$

### 2.3.0.5 Função Rosenbrock

A função Rosenbrock também é conhecida como função Valley ou Banana. O gráfico, Figura 11 mostra sua forma bidimensional. E sua expressão é apresentada na Equação 2.10, onde  $d$  é o tamanho da dimensão,  $x_i \in [-5, 10]$ , embora isso possa restringir ao hiper-cubo  $x_i \in [-2.048, 2.048]$ , para todo  $i = 1, \dots, d$  e seu mínimo global é  $f(x^*) = 0$ , em  $x^* = (1, \dots, 1)$ .

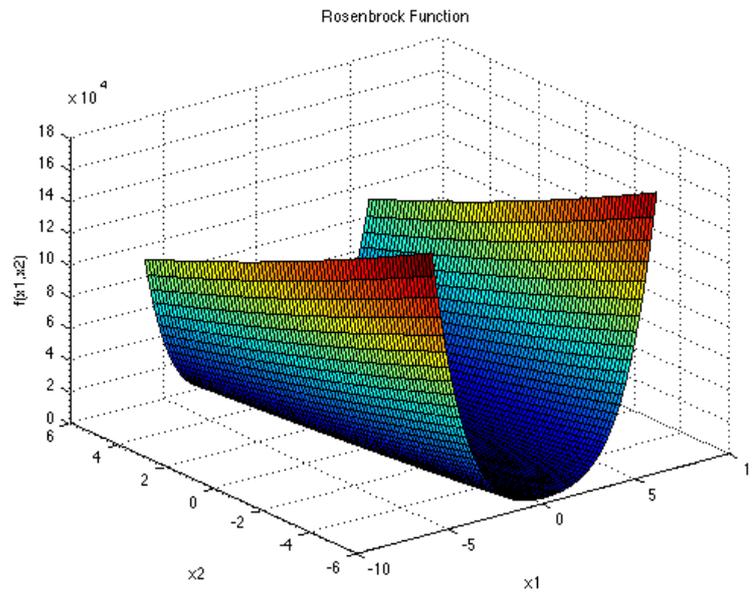
Figura 10 – Gráfico da função Dixon Price



Fonte: (SURJANOVIC; BINGHAM, 2015)

$$f(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (2.10)$$

Figura 11 – Gráfico da função Rosenbrock



Fonte: (SURJANOVIC; BINGHAM, 2015)

### 2.3.0.6 Função Levy

A função Levy é ilustrada na Figura 12. Sua expressão é apresentada na Equação 2.11, onde  $d$  é o tamanho da dimensão,  $x_i \in [-10, 10]$ , para todo  $i = 1, \dots, d$  e seu mínimo

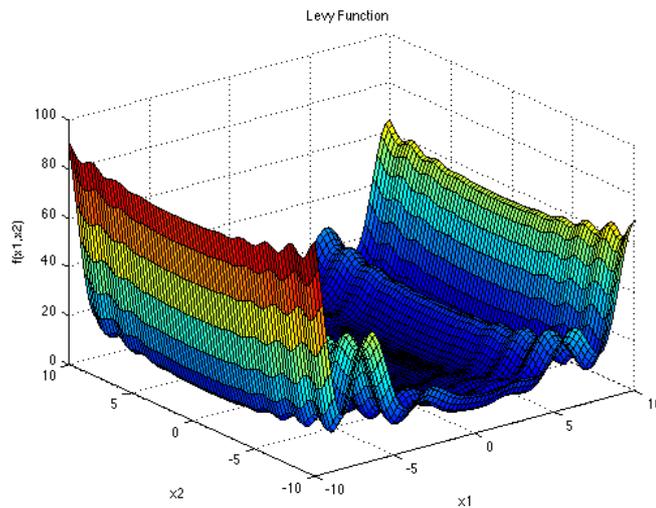
global é  $f(x^*) = 0$ ,  $x^* = (1, \dots, 1)$ .

$$f(x) = \sin^2(\pi w_1) + \sum_{i=1}^{d-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1)] + (w_d - 1)^2 [1 + \sin^2(2\pi w_d)] \quad (2.11)$$

onde

$$w_i = 1 + \frac{x_i - 1}{4} \quad (2.12)$$

Figura 12 – Gráfico da função Levy



Fonte: (SURJANOVIC; BINGHAM, 2015)

## 2.4 Teste de Wilcoxon pareado

O Wilcoxon pareado foi criado por Frank Wilcoxon em 1945 (WILCOXON, 1945). É um teste não-paramétrico de hipótese estatística utilizado para comparar duas amostras relacionadas, amostras pareadas, ou medições repetidas em uma única amostra. Além disto, é usada para avaliar o quanto as amostras são idênticas estatisticamente ou se diferem (WILCOXON; KATTI; WILCOX, 1970).

O teste considera duas amostras ( $A_1$  e  $A_2$ ) de tamanho  $n$  ( $x_1, \dots, x_n$  e  $y_1, \dots, y_n$ ). Por se tratar de observações pareadas, pode-se considerar uma formação de pares das duas amostras, um valor oriundo da primeira amostra,  $A_1$ , e o outro da segunda amostra,  $A_2$ . A partir dos pares formados, calcula-se a diferença ( $D_i = x_i - y_i$ , para todo  $i = 1, 2, \dots, n$ ) de cada par e obtém-se uma terceira amostra,  $Dif$  (ESTATCAMP, 2005c).

Na Tabela 1, segue um exemplo de duas amostras aplicadas no teste de Wilcoxon.

Para realizar o Teste de Wilcoxon pareado deve-se estabelecer as seguintes hipóteses:

Tabela 1 – Exemplo de amostras de teste Wilcoxon pareado

	1	2	3	4	5	6
A1	52	15	26	44	12	10
A2	49	17	22	39	19	11
<i>Dif</i>	3	-2	4	5	-7	-1

Fonte: o autor

- $\begin{cases} H_0 : \Delta = 0 \\ H_1 : \Delta \neq 0 \end{cases}$
- $\begin{cases} H_0 : \Delta = 0 \\ H_1 : \Delta > 0 \end{cases}$
- $\begin{cases} H_0 : \Delta = 0 \\ H_1 : \Delta < 0 \end{cases}$

Onde:

- $H_0$  é a hipótese nula;
- $H_1$  é a hipótese alternativa e;
- $\Delta$  é a mediana das diferenças entre os pares das amostras.

Dessa forma é testado se as amostras se diferem ou não: se a hipótese nula satisfizer, a mediana da diferença é nula, ou seja, as amostras não diferem. Caso contrário, se a hipótese nula não satisfizer, a mediana da diferença não é nula, ou seja, tem que as amostras se diferem (ESTATCAMP, 2005c).

Uma vez calculadas todas as diferenças entre os valores obtidos para cada par de dados da amostra e levantadas as hipóteses, o critério para o teste, definido por Wilcoxon, Katti e Wilcox (1970), é calculado da seguinte maneira: primeiro são ordenadas de forma crescente a *Dif* por seus valores absolutos. Feito isso, é definido um indicador ( $c_i$ ) de valor positivo ou negativo, onde  $c_i = 0$  se a *Dif* do índice  $i$  for menor que zero, e  $c_i = 1$  caso contrário, exemplo na Tabela 2. Por fim, é feito o somatório das diferenças positivas utilizando a Equação 2.13.

$$W = \sum_{i=1}^n ic_i \quad (2.13)$$

*i.e.*,  $W$  é a soma das diferenças positivas e  $n$  é o tamanho da amostra. Aplicado no exemplo o seguinte resultado:

Tabela 2 – Diferenças ordenadas pelo seu valor absoluto.

$ Dif $	-1	-2	3	4	-5	-7
$i$	1	2	3	4	5	6
$c_i$	0	0	1	1	0	0

Fonte: o autor

$$W = 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 1 + 4 \cdot 1 + 5 \cdot 0 + 6 \cdot 0 = 3 + 4 + 5 = 7$$

Com o auxílio do software *R-project* é possível realizar o teste de Wilcoxon pareado. Tal ferramenta é um software livre e considerado como um ambiente para computação estatística e gráficos (R Core Team, 2016).

Para proceder o teste no *R-project*, as amostras devem ser lidas como vetores, conforme no exemplo abaixo:

***a1 <- c(52.0 , 15.0, 26.0, 44.0, 12.0, 10.0)***

***a2 <- c(49.0 , 17.0, 22.0, 39.0, 19.0, 11.0)***

Após passadas as amostras para o software, pode-se executar o método para o teste de Wilcoxon pareado no *R-project*. Como se trata de amostras pareadas, o objeto *paired* deve ser definido como verdadeiro, ou seja atribuído com valor *true*. Logo, a sintaxe do comando é *wilcox.test(a1, a2, paired = TRUE)*. O comando retornará o valor da estatística  $V$ , que é o somatório das diferenças positivas ( $W$ ); e o *p-value* (*p-valor*), o qual indica se as amostras possuem ou não diferenças significativas. A sintaxe do método dado o exemplo anterior é o seguinte:

***Wilcoxon signed rank test***

***data: a1 and a2***

***V = 12, p-value = 0.8438***

***alternative hypothesis: true location shift is not equal to 0.***

A eficiência do teste de Wilcoxon é de cerca de 95% tanto para pequenas como para grande amostras. Isto significa que, se o teste retornar um valor superior a 5%, as amostras não apresentam diferenças significativas. Caso contrário, se o teste resultar um valor menor que 5%, as amostras são estatisticamente diferentes.

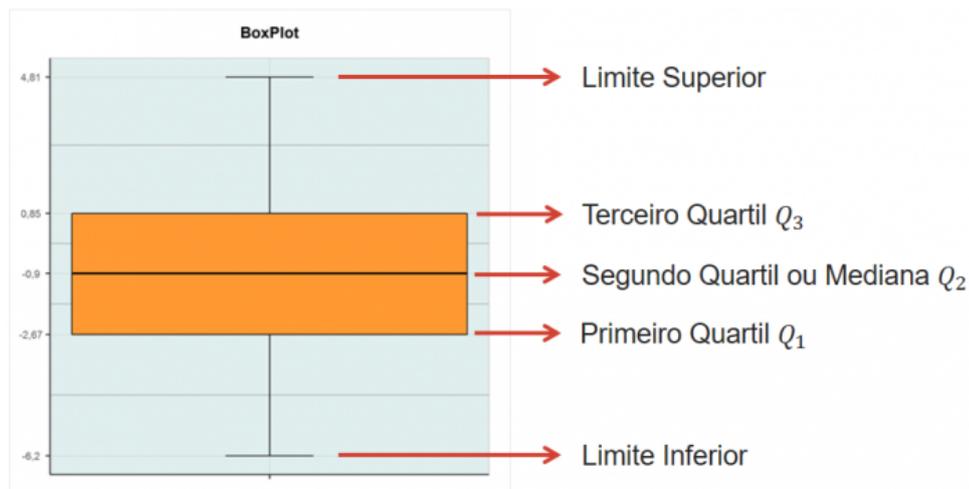
Como *p-value* = 0,8438 ( $> 0,05$ ), é possível afirmar que as medianas das amostras não estão consideravelmente distantes e não apresentam diferenças significativas.

Para caso o retorno do *p-value* ser um valor menor que 0,05, e assim descobrir qual das amostras gerou uma melhor população, calcula-se as medianas das mesmas. Para maximizar os resultados, a mediana com o maior valor indicará a resposta mais adequada e se o caso for para minimizar, a mediana com o menor valor é considerada a melhor amostra (GONÇALVES, 2016). O cálculo da mediana das amostras pelo software *R-project*, é feito

pelo comando `summary(x)`, sendo  $x$  o vetor da amostra passada por parâmetro.

A fim de representação gráfica das amostras, pode-se utilizar um *boxplot*, que é formado pelo primeiro, terceiro quartil e pela mediana (ESTATCAMP, 2005a) (Figura 13). Quartis são valores representados a partir do conjunto ordenado de observações de forma crescente, que dividem a distribuição em quatro partes iguais (ESTATCAMP, 2005b). As hastes inferiores e superiores se estendem, respectivamente, do quartil inferior e do quartil superior.

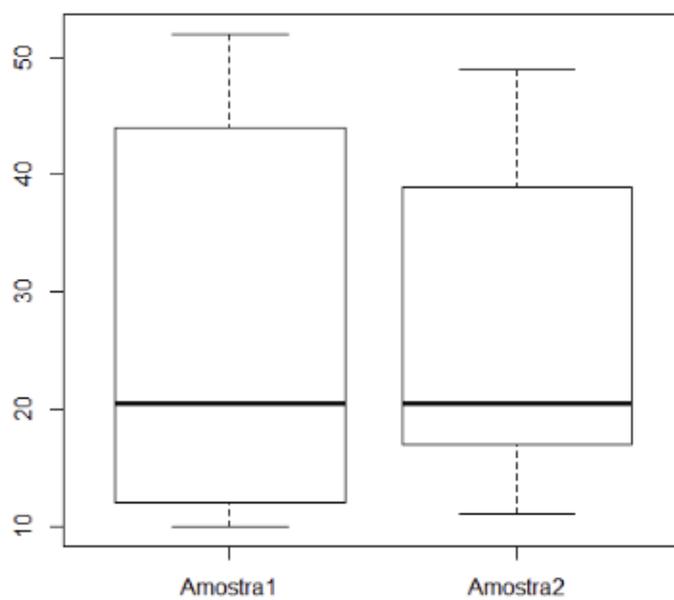
Figura 13 – Representação gráfica do *boxplot*



Fonte: (ESTATCAMP, 2005a)

Para plotagem do gráfico *boxplot*, seguindo o exemplo anterior, utiliza-se o comando adiante, como resultado é representado na Figura 14:

```
boxplot(a1, a2, names = c("Amostra1", "Amostra2"))
```

Figura 14 – Representação gráfica do *boxplot* para o exemplo

Fonte: o autor

### 3 Procedimentos metodológicos

Para o desenvolvimento deste trabalho foi tomado como base uma sequência de etapas que envolveram desde o estudo e compreensão dos conceitos de AG até o desenvolvimento e adaptação do mesmo para o alcance do objetivo proposto.

Os procedimentos metodológicos são organizados nas seguintes etapas:

1. O levantamento bibliográfico começou em um estudo sobre uma adaptação de convergência para o AG e seguiu por pesquisar em alguns repositórios científicos online, tais como: *IEEE Explorer*, Google Acadêmico, Vértice, Centros Acadêmicos. Os termos principais para a pesquisa foram: algoritmo genético, *genetic algorithm*, *rechenberg*, regra um quinto de sucesso e entre outros;
2. Para conceber um melhor estudo e obtenção de resultados foi utilizada as funções de *benchmark* (SURJANOVIC; BINGHAM, 2015), pois as mesmas são utilizadas para problemas de otimização. Neste caso as funções de  $d$  dimensões foram as escolhidas devido a situação do problema proposto;
3. O AG utilizado neste trabalho, foi o mesmo da proposta de Paula Peçanha Gonçalves (GONÇALVES, 2016), definido como FGA;
4. A implementação da regra de um quinto de sucesso, elaborada por Rechenberg (1973) foi descrita também por Linden (2012) em seu livro. Tomado o conceito da mesma, foi feita uma adaptação no operador de *crossover* do FGA e assim o algoritmo proceder com parâmetro de evolução, gerando o FGA+RS;
5. A execução do AG, submetido às funções de *benchmark*, procedeu em dois modos, com e sem a regra de um quinto de sucesso aplicado no operador de *crossover*;
6. Cada execução do AG implica em evoluir as soluções candidatas por  $n$  gerações e na última, obter aquela que é a melhor. A fim de análise a implementação será executada 25 vezes, obtendo os 25 melhores indivíduos das execuções (amostras). Desta forma foi apresentado o menor e maior valor, valor médio e desvio padrão das amostras;
7. O teste de Wilcoxon pareado foi utilizado para apresentar, estatisticamente, o quanto as amostras se diferem, caso positivo;
8. E por fim, a análise dos resultados no aspecto da viabilidade da implantação da combinação da regra de um quinto de sucesso aplicado no operador de *crossover*, em relação à genérica.

## 4 Implementação

Neste capítulo está descrito a forma que foi implementado a proposta deste trabalho, a modelagem da população e os operadores genéticos. Assim como, as duas implementações de AG.

### 4.1 Implementação dos AGs

A implementação dos algoritmos foi feita utilizando o ambiente de desenvolvimento NetBeans 8.1 criado pela empresa Sun Microsystems, no sistema operacional Windows 7 Home, 64-bit. O computador utilizado possui um processador Core i3 2.00 GHz; 4,00 GB de memória RAM; e 1 TB de espaço de armazenamento. Foi escolhida a linguagem de programação Java como sugerida no livro do Linden (2012).

Nesta proposta, foram utilizadas as funções mencionadas na Seção 2.3 para avaliar os indivíduos e poder comparar o desempenho do FGA e FGA+RS. Para cada função, é gerada uma população inicial, respeitando o seu intervalo. A mesma população inicial é o ponto de partida dos dois algoritmos, que ao final das de gerações, retorna uma população final.

O programa foi executado 25 vezes para cada AG implementado, gerando assim 25 amostras. O tamanho da população e dimensão dos indivíduos são fixos, para o primeiro teste, foi definido 100 indivíduos e 10 genes; e no segundo teste, 200 indivíduos e 20 genes. Além disto, a quantidade de gerações a ser percorrida é de 300 gerações, para os dois testes realizados.

Tratando de um estudo de adaptação na primeira implementação, os processos genéticos de mutação, recombinação e seleção são os mesmos. A principal diferença, para segunda implementação, é a taxa de corte do operador de *crossover*, a qual sofre uma alteração conforme o índice de melhoras dos filhos em relação ao seus pais.

A representação dos alelos utilizada foi de números reais inerentes ao intervalo da função de avaliação.

Os métodos de seleção foram: em cada geração, 20% dos indivíduos são selecionados por elitismo. Ou seja, os 20% melhores indivíduos de cada geração são passados para a próxima geração. Como as funções de avaliação utilizadas são de minimização, os indivíduos são ordenados de forma decrescente, considerando o retorno da função de avaliação de menor valor como o melhor valor; e os 80% restantes são selecionados pelo método da roleta viciada (*roulette wheel*), como já visto na Seção 2.1.4, o mesmo permite que todos os indivíduos tenham chance de ser selecionado para próxima geração, inclusive aquele que

não for classificado como um bom candidato. Entretanto, a probabilidade de um indivíduo ser selecionado é proporcional a seu *fitness*, ou seja, quanto melhor o *fitness*, maior a chance do indivíduo ser escolhido.

Para problemas de minimização, que é o caso deste trabalho, foi utilizado a roleta viciada com o retorno da função de avaliação inversamente proporcional, ou seja, o menor valor do resultado, ocupará um espaço maior na mesma.

No processo de recombinação foi escolhido o operador *crossover* aritmético, por utilizar valores reais para representação dos indivíduos. Nesta estratégia, dois progenitores aleatórios geram dois descendentes. A taxa de proporção dos pais usada é definida por uma variável aleatória uniforme lambda ( $\lambda \in [0, 1]$ ).

Neste processo, há uma diferença entre a implementação original e adaptada, que se dá na quantidade de genes recombinados.

Na implementação FGA, foi utilizado a recombinação em todos os genes de cada indivíduo.

No FGA+RS, a quantidade de genes recombinados inicialmente é o tamanho do indivíduo. E por se tratar de uma estratégia evolutiva, essa taxa de genes é alterada da seguinte forma: se o índice melhora for menor que um quinto, logo está próximo de um mínimo local e deve incrementar um gene; caso contrário, se o índice melhora for maior que um quinto, logo estou distante de um máximo local e deve decrementar um gene. Desta forma o AG adaptado passa-se pelos processos de evolução durante as gerações.

A estratégia para o processo de mutação, foi a mutação gaussiana. Nesta estratégia, é sorteado um gene aleatório do indivíduo para sofrer uma alteração e a probabilidade de mutação escolhida é o valor real aleatório entre 0 e 1. Depois de selecionado um gene pela probabilidade é gerado um número aleatório pela distribuição normal. Tal número é usado para realizar a perturbação (mutação) desse gene da seguinte forma: o valor do gene é somado com este número. A ideia de aplicar a mutação nos algoritmos genéticos, possibilita explorar por outras soluções e sair de máximos ou mínimos locais.

### 4.1.1 Implementação FGA

O modelo do AG utilizado neste trabalho foi proposto por Douglas Nunes de Oliveira e Paula Peçanha Gonçalves (GONÇALVES, 2016), definido como FGA (*full genetic algorithm*). Neste AG a competição pela sobrevivência está entre pais, filhos e mutantes.

Nesta implementação o algoritmo gera uma população inicial denominada como pai. Esta população é preenchida com valores aleatórios dentro do intervalo de cada função *benchmark*. A partir dela são geradas mais duas: a população de filhos e a população de

mutantes. A população de filhos é formada no processo de recombinação pelo operador de *crossover* aritmético. E todos os indivíduos da população de mutantes sofrem alteração em um gene aleatório. A avaliação de cada população é feita no final do processo de criação da mesma.

Posteriormente as três populações são concatenadas, formando assim uma população com o número de indivíduos três vezes maior que a população inicial.

A população para a próxima geração possui a mesma dimensão da população inicial. Para ser gerada, é feita a ordenação de forma decrescente pelo *fitness* (dos mais aptos aos menos aptos) da população concatenada. Após a concatenação, é selecionado da mesma 20% dos indivíduos por elitismo em relação da população inicial, ou seja, no exemplo do primeiro teste utilizou-se 100 indivíduos, logo deve selecionar 20 indivíduos da população concatenada. E os 80 indivíduos restante da população futura são preenchidos pelo método *roulette wheel*.

Esta população é alterada de geração a geração, formando novos filhos e mutantes até que alcance a quantidade de gerações definida para o teste.

O procedimento desta implementação pode ser representado pelo fluxograma na Figura 15 e seu Pseudocódigo 1.

---

#### Algorithm 1 Pseudocódigo FGA

---

```

1: Definir valores das variáveis: quantidadeDeFunções e quantidadeDeGerações;
2: for função_n do 0 até quantidadeDeFunções
3:   Gerar população inicial com valores aleatórios no intervalo da função corrente
   (função_n);
4:   Avaliar população inicial com a função corrente (função_n);
5:   for geração do 1 até quantidadeDeGerações
6:     Gerar população de filhos por recombinação crossover aritmético;
7:     Avaliar população de filhos com a função corrente;
8:     Gerar população de mutante por mutação gaussiana;
9:     Avaliar população de mutantes com a função corrente;
10:    Concatenar: população inicial + população de filhos + população mutados;
11:    Ordenar população concatenada;
12:    Selecionar 20% dos indivíduos por elitismo, proporcional a população inicial;
13:    Selecionar 80% pelo método roulette wheel, proporcional a população inicial;
14:    Substituir na população inicial os indivíduos selecionados;
15:  end for
16: end for

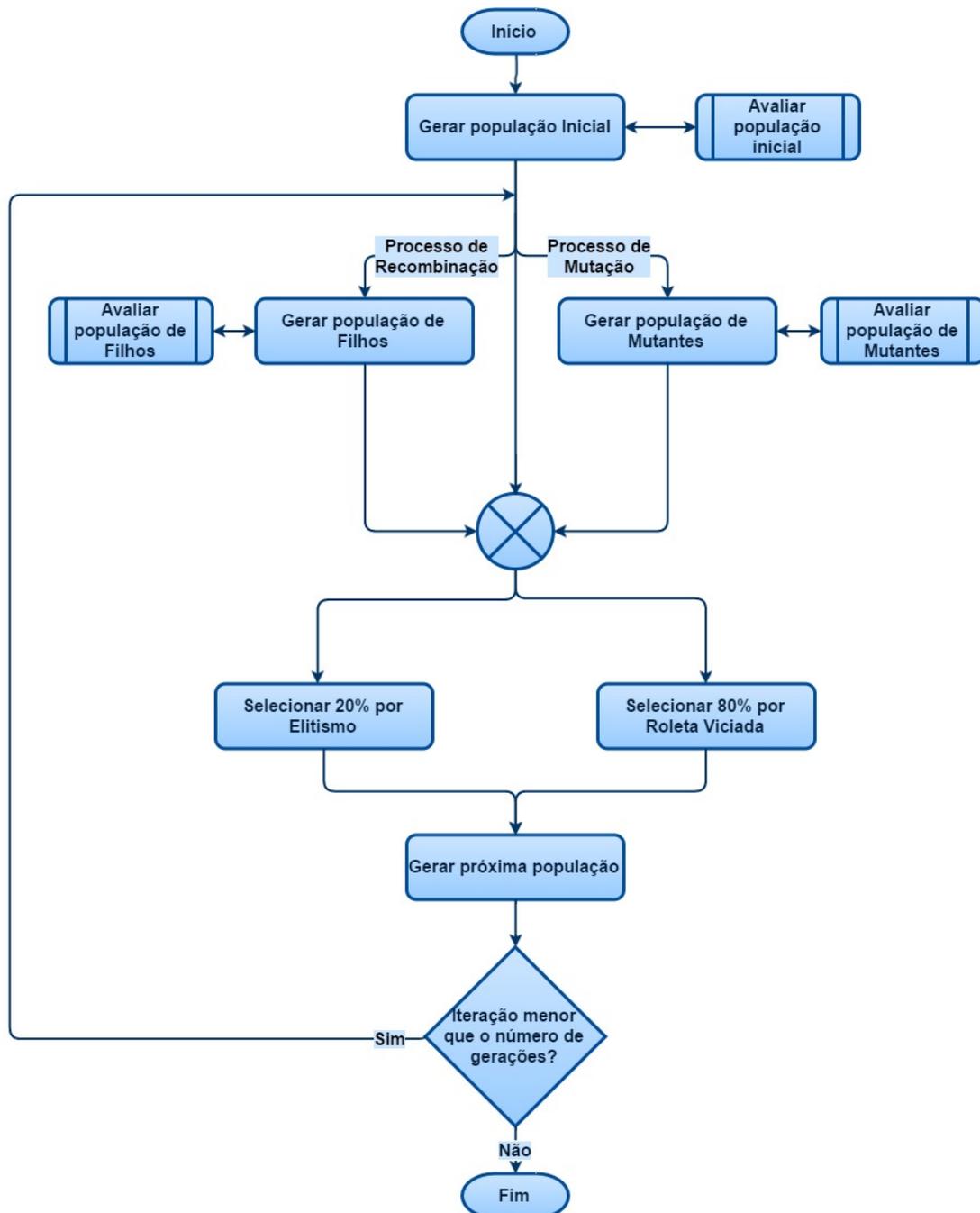
```

---

### 4.1.2 Implementação FGA+RS

O significado de RS é "regra de sucesso" devido ao nome da estratégia utilizada. Esta implementação foi feita adaptando a regra um quinto de sucesso no operador de recombinação do FGA.

Figura 15 – Fluxograma do FGA



Fonte: o autor

Conhecida a proposta deste trabalho, que se trata de uma investigação do uso da regra de um quinto de sucesso no operador de *crossover* durante o processo de recombinação. Foi necessário incluir a estratégia de um quinto de sucesso (RECHENBERG, 1973) na recombinação para proceder a adaptação, para que o operador de *crossover* pudesse trabalhar como o parâmetro de evolução.

A adaptação desta implementação, partiu da ideia que "um quinto dos filhos gerados devem ser melhores do que seus pais." (LINDEN, 2012). Posto isto, a cada  $n$  gerações,

ao longo do processo de busca, calcula o índice de melhora, *i. e.*, a razão do número de sucessos e o número de iterações. Vale ressaltar que o tamanho do indivíduo (dimensão) é fixo e a quantidade de genes a variar é a proporção de genes que irá ser aplicado no operador de *crossover*. Então, foi adaptada da seguinte forma: se o índice de melhora for maior que um quinto, decrementa uma unidade na quantidade genes aplicado no operador de *crossover*; se o índice de melhor for menor que um quinto incrementa uma unidade na quantidade genes aplicado no operador de *crossover*.

Assumindo a aplicação desta regra periodicamente em todas as execuções, a sua expressão, Equação 4.1 pode ser definida da seguinte forma para uma iteração:

$$\sigma = \begin{cases} \sigma, p_{melhora} = 1/5 \\ \sigma - 1, p_{melhora} > 1/5 \\ \sigma + 1, p_{melhora} < 1/5 \end{cases} \quad (4.1)$$

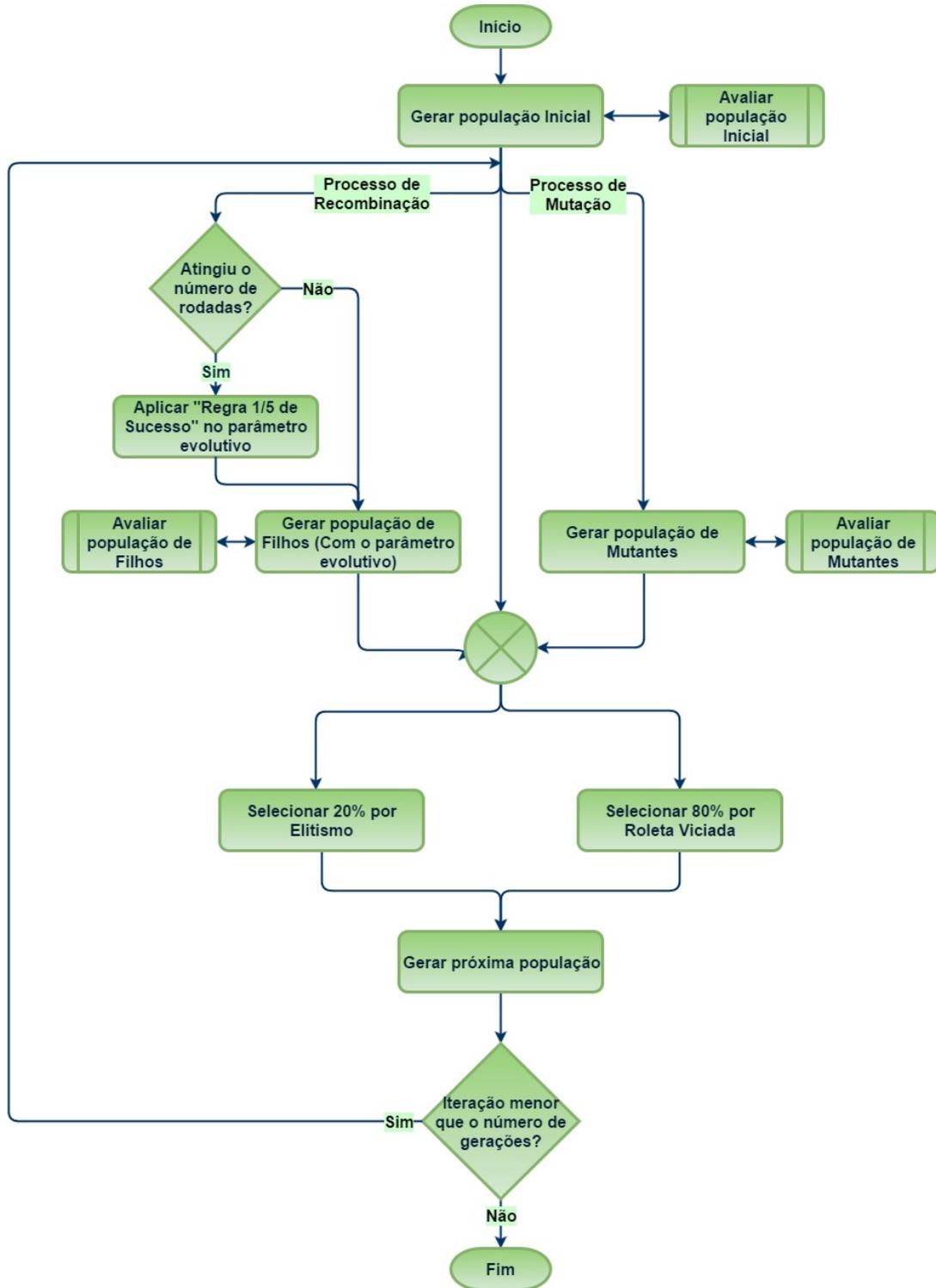
Onde:

- $\sigma$  é o parâmetro de evolução;
- $p_{melhora}$  é o índice de melhora nas últimas  $n$  gerações.

Desta forma, a execução do FGA+RS é feita com todas as funções objetivo, conforme no FGA. E em ambas implementações os demais processos de seleção e mutação.

O procedimento desta implementação pode ser representado pelo fluxograma na Figura 16 e seu Pseudocódigo 2.

Figura 16 – Fluxograma do FGA+RS



Fonte: o autor

---

**Algorithm 2** Pseudocódigo FGA+RS

---

```
1: Definir valores das variáveis: quantidadeDeFunções, quantidadeDeGerações e
   numeroDeRodadas;
2: for função_n do 0 até quantidadeDeFunções
3:   Gerar população inicial com valores aleatórios no intervalo da função corrente
   (função_n);
4:   Avaliar população inicial com a função corrente (função_n);
5:   for geração do 1 até quantidadeDeGerações
6:     Gerar população de filhos por recombinação crossover aritmético;
7:     Avaliar população de filhos com a função corrente;
8:     if (geração%numeroDeRodadas = 0) and (geração > 0) then
9:       if (taxaDeMelhora > 1/5) then
10:        Decrementar quantidade de genes recombinados;
11:       end if
12:       if (taxaDeMelhora < 1/5) then
13:        Incrementar quantidade de genes recombinados;
14:       end if
15:     end if
16:     Gerar população de mutante por mutação gaussiana;
17:     Avaliar população de mutantes com a função corrente;
18:     Concatenar: população inicial + população de filhos + população mutados;
19:     Ordenar população concatenada;
20:     Selecionar 20% dos indivíduos por elitismo, proporcional a população inicial;
21:     Selecionar 80% pelo método roulette wheel, proporcional a população inicial;
22:     Substituir na população inicial os indivíduos selecionados;
23:   end for
24: end for
```

---

## 5 Resultados

Os resultados apresentados e analisados nesta seção, tratam-se dos *fitness* resultantes das funções objetivo utilizadas, a fim de comparar as implementações propostas neste trabalho, investigando então, o uso da regra de um quinto de sucesso aplicado no operador de *crossover*.

### 5.1 Apresentação dos resultados

Conforme já demonstrado no [Capítulo 4](#), as duas implementações de AG foram submetidas a 25 execuções. Para cada execução, foi criada uma população inicial e avaliada pelas seis funções objetivo utilizadas, gerando uma população final no decorrer das gerações. No final de cada execução, foi coletado o melhor indivíduo (aquele que possui melhor *fitness*) de todos os indivíduos. E se tratando de um problema de minimização, o melhor valor avaliado é o menor valor retornado pela função objetivo.

#### 5.1.1 Primeiro teste

O primeiro teste feito, utilizou os seguintes parâmetros: populações com 100 indivíduos com 10 genes e 300 gerações. A partir do resultado do melhor indivíduo encontrado, foram elaboradas tabelas para apresentação dos dados, organizados por cada função objetivo e implementação. Essas tabelas se encontra na seção de Apêndices ([Apêndice A](#)) e foram resumidas na Tabela 3.

Tabela 3 – Resumo do primeiro teste

Função	Implementação	Menor valor	Média	Maior valor	Desvio Padrão
Griewank	FGA	1,852E-01	1,089E+00	2,x+00	5,790E-01
	FGA+RS	2,666E-01	1,006E+00	2,592E+00	1,294E+01
Rastringin	FGA	5,633E-03	1,906E+00	4,082E+00	1,342E+00
	FGA+RS	5,497E-03	1,529E+00	4,800E+00	5,607E+01
Schwefel	FGA	3,498E+03	3,744E+03	3,915E+03	8,200E+01
	FGA+RS	3,149E+03	3,719E+03	3,936E+03	7,297E+05
Dixon-Price	FGA	5,001E-01	5,008E-01	5,026E-01	1,106E-03
	FGA+RS	4,455E-01	4,983E-01	5,015E-01	2,589E-06
Rosenbrock	FGA	3,823E-01	1,128E+00	5,390E+00	3,224E+00
	FGA+RS	3,055E-01	3,149E+00	5,186E+01	4,298E+01
Levy	FGA	0,000E+00	3,799E-21	9,497E-20	0,000E+00
	FGA+RS	0,000E+00	3,425E-06	8,562E-05	0,000E+00

Fonte: o autor

A Tabela 3 é composta por:

- Função: nome da função aplicada à implementação;
- Menor valor: o menor valor retornado pela função Objetivo (maior *fitness*) dentre as 25 amostras;
- Média: a média do valor retornado pela função Objetivo das 25 amostras, calculado pela média aritmética;
- Maior valor: o maior valor retornado pela função Objetivo (menor *fitness*) encontrado dentre as 25 amostras;
- Desvio Padrão: a dispersão das 25 amostras em relação ao valor médio amostral.

Baseado nos mesmos dados coletados (Apêndice A) e resumidos na Tabela 3, foi aplicado o teste de Wilcoxon pareado para analisar estatisticamente os mesmos. A Tabela 4, apresenta as informações aplicadas ao teste via software *R-project*.

Tabela 4 – Resultados do primeiro experimento no teste Wilcoxon pareado

Função	Estatística	<i>p-value</i>
Griewank	153	8,119E-01
Rastrigin	146	6,721E-01
Schwefel	219	1,336E-01
<b>Dixon-Price</b>	<b>243</b>	<b>2,958E-02</b>
Rosenbrock	175	7,510E-01
Levy	5	4,227E-01

Fonte: o autor

Os resultados pelo teste de Wilcoxon pareado, Tabela 4, apenas a função Dixon-Price retornou um *p-value* menor que 0.05, o que indica que os valores das amostras, para esta função, são estatisticamente diferentes. No entanto, as demais funções denotam que as amostras não apresentam diferenças significativas.

No caso da função que retornou um *p-value* menor que 0.05, foi calculado a mediana, para descobrir qual das amostras gerou uma melhor população. Como o propósito da pesquisa é minimizar o valor, a menor mediana é quem apresenta a melhor amostra.

Para determinar qual das duas implementações demonstrou melhor comportamento para a função Dixon-Price, a qual *p-value* foi menor que 0.05, calculou-se as medianas das amostras. Em destaque na Tabela 5, a mesma resultou uma mediana menor para a implementação FGA+RS, e portanto, uma melhor eficiência posto que é desejava minimizar os resultados.

Tabela 5 – Mediana para primeiro teste

Função	Implementação	Mediana
Griewank	FGA	7,227E-01
	FGA+RS	8,041E-01
Rastrigin	FGA	1,375E+00
	FGA+RS	2,000E+00
Schwefel	FGA	3,776E+03
	FGA+RS	3,731E+03
<b>Dixon-Price</b>	<b>FGA</b>	<b>5,007E-01</b>
	<b>FGA+RS</b>	<b>5,004E-01</b>
Rosenbrock	FGA	1,053E+00
	FGA+RS	1,149E+00
Levy	FGA	0,000E+00
	FGA+RS	0,000E+00

Fonte: o autor

Para apresentar as amostras graficamente, as estatísticas dessas foram plotadas no *R-project* por *boxplot* e são mostradas na Figura 17.

### 5.1.2 Segundo teste

Para comparar os resultados obtidos no primeiro teste um segundo teste foi feito. Neste alterou-se o tamanho das populações. E diferentemente do anterior, este experimento possui população de 200 indivíduos com 20 genes.

Os resultados obtidos por cada função se encontram no [Apêndice B](#). Os mesmos foram resumidos e apresentados na Tabela 6. Destacando o menor valor, a média e o maior valor obtido pelas funções por implementação.

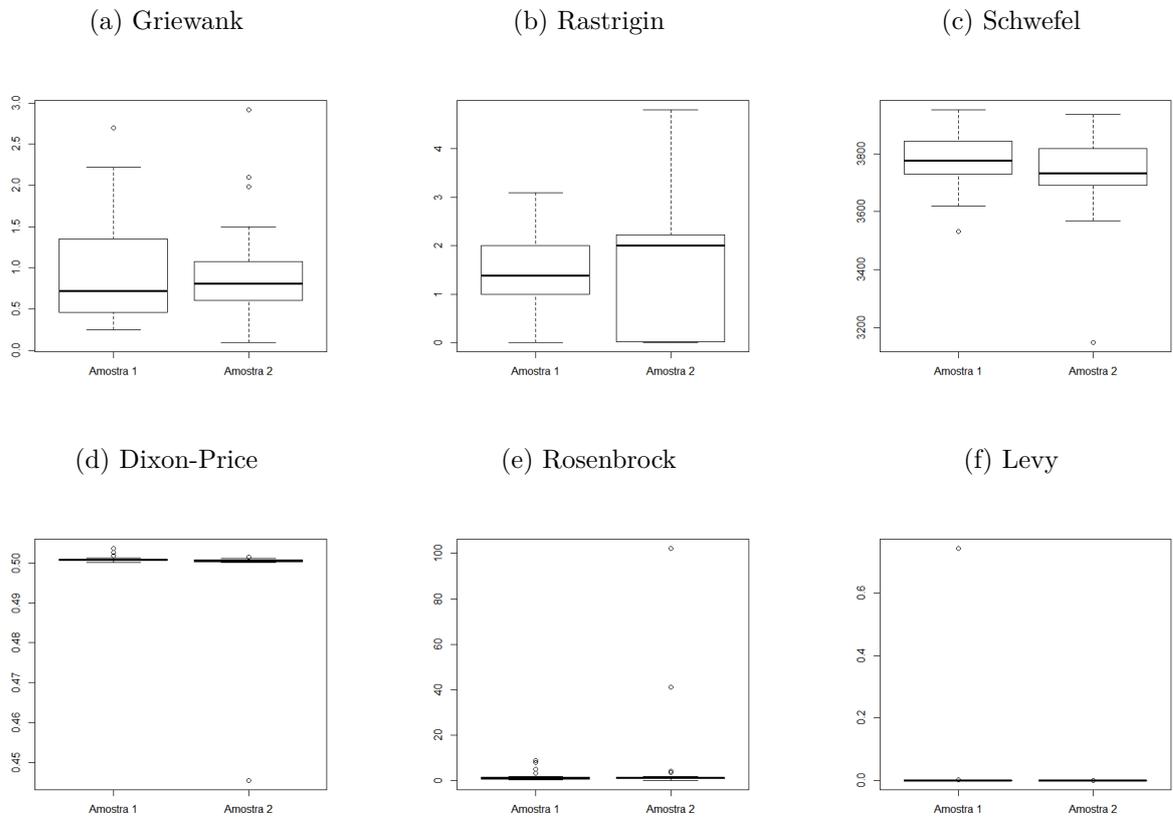
A Tabela 7, apresenta as respostas obtidas pelo teste de Wilcoxon pareado para análise das populações. Nessa tabela, todas as medidas de *p-value* encontradas são maiores que 0.05. Sendo assim, pôde-se conferir que as amostras não possuem diferenças estatisticamente. No caso da função Levy, que não resultou o teste de Wilcoxon, foi devido todos os seus dados amostrais ter sido zero.

Para apresentar as amostras graficamente, as estatísticas dessas foram plotadas no *R-project* por *boxplot* e são mostradas na Figura 18.

### 5.1.3 Tempo de execução

O tempo de execução de um programa é levado em consideração para descobrir o desempenho e o tempo de resposta do mesmo. Por este motivo, foi dado a importância também o ao tempo de execução de cada implementação.

Figura 17 – *Boxplots* do *R-project* para o primeiro teste



Fonte: o autor

Tabela 6 – Resumo do segundo teste

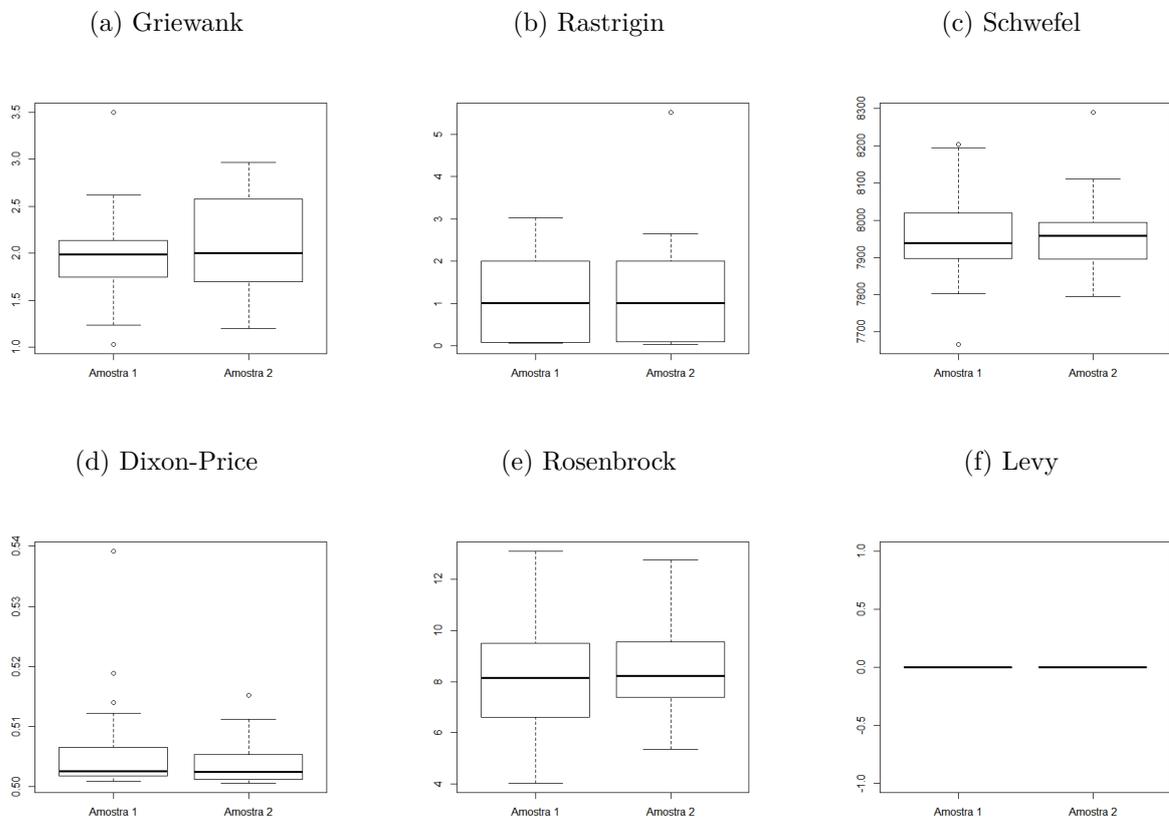
Função	Implementação	Mínimo	Média	Máximo	Desvio Padrão
Griewank	FGA	1,032E+00	1,976E+00	3,497E+00	5,790E-01
	FGA+RS	1,197E+00	2,087E+00	2,963E+00	1,579E+00
Rastrigin	FGA	6,219E-02	1,041E+00	3,021E+00	9,545E-01
	FGA+RS	3,642E-02	1,185E+00	5,511E+00	1,200E+00
Schwefel	FGA	7,667E+03	7,963E+03	8,204E+03	1,255E+02
	FGA+RS	7,795E+03	7,957E+03	8,290E+03	1,015E+02
Dixon-Price	FGA	5,009E-01	5,060E-01	5,391E-01	8,276E-03
	FGA+RS	5,006E-01	5,037E-01	5,152E-01	3,625E-03
Rosenbrock	FGA	4,024E+00	8,036E+00	1,311E+01	2,123E+00
	FGA+RS	5,351E+00	8,400E+00	1,278E+01	1,948E+00
Levy	FGA	0,000E+00	0,000E+00	0,000E+00	0,000E+00
	FGA+RS	0,000E+00	0,000E+00	0,000E+00	0,000E+00

Fonte: o autor

Tabela 7 – Resultados do segundo experimento no teste Wilcoxon pareado

Função	Estatística	<i>p-value</i>
Griewank	144	6,338E-01
Rastrigin	164	9,789E-01
Schwefel	169	8,740E-01
Dixon-Price	188	5,077E-01
Rosenbrock	143	6,150E-01
Levy	0	NA

Fonte: o autor

Figura 18 – *Boxplots* do *R-project* para o segundo teste

Fonte: o autor

Para os dois testes realizados e descritos na seção 5.1, foram coletados o tempo de execução de cada implementação. A Tabela 8 foi elaborada para apresentar os resultados dos tempos de execução.

Nos testes executados (Tabela 8) há uma pequena diferença no tempo de execução das implementações. Tal diferença denotada, reflete que não há muita discrepância na eficiência da execução das implementações.

Tabela 8 – Tempo de execução dos testes

Teste	Implementação	Tempo (s)
Primeiro teste	FGA	144,122
	FGA+RS	140,485
Segundo teste	FGA	511,167
	FGA+RS	511,670

Fonte: o autor

Para demais comparações do tempo de execução entre as implementações, outros testes foram feitos. Deste modo, a dimensão dos indivíduos foram fixados com 10 genes e o tamanho da população foi variado para 100, 200, 300, 400 e 500 indivíduos. Foi elaborada a Tabela 9 para apresentação dos tempos.

Tabela 9 – Tempo de execução para tamanho de população diferentes

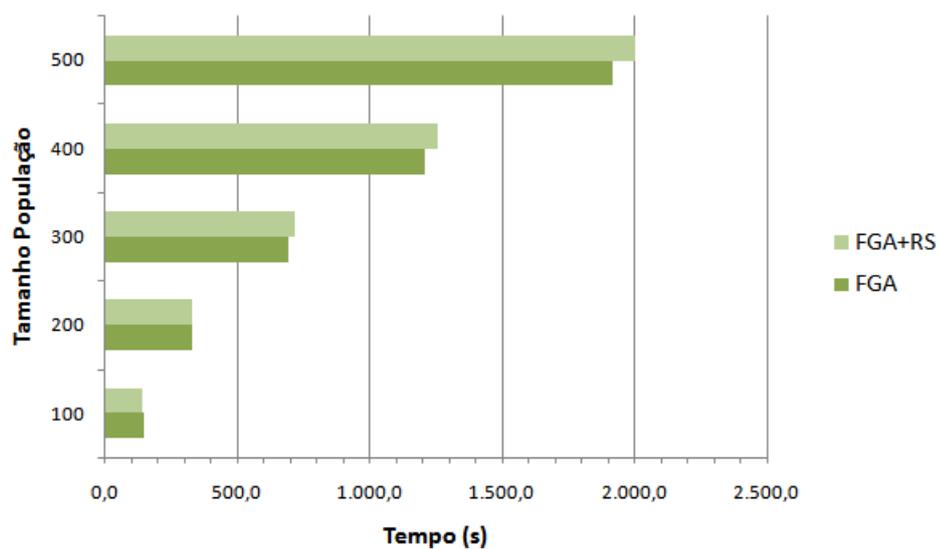
Parâmetros		Implementação	Tempo (s)
Dimensão: 10 genes	População: 100 indivíduos	FGA	144,122
		FGA+RS	140,485
	População: 200 indivíduos	FGA	327,007
		FGA+RS	327,483
	População: 300 indivíduos	FGA	693,857
		FGA+RS	714,985
	População: 400 indivíduos	FGA	1206,029
		FGA+RS	1254,766
	População: 500 indivíduos	FGA	1917,899
		FGA+RS	2004,849

Fonte: o autor

Nota-se que o tempo de execução da implementação FGA foi menor que da implementação FGA+RS, na maioria dos testes. Apesar da pequena diferença do tempo de execução, isso ocorreu pelo fato da implementação FGA+RS aplicar a regra de um quinto de sucesso durante o seu processo de evolução. Além disso, o tempo de execução aumenta proporcionalmente ao tamanho da população.

Foi elaborado o gráfico na Figura 19 para melhor visualizar os tempos de execução, com as populações de tamanho 100 a 500 indivíduos e dimensão de 10 genes.

Figura 19 – Gráfico dos tempos de execução



Fonte: o autor

## 6 Conclusão

A partir dos dois testes realizados, foi considerado os parâmetros de menor valor, média, maior valor, desvio padrão e o teste de Wilcoxon pareado, verificou-se o seguinte:

- no primeiro teste, a implementação FGA+RS apresentou um melhor resultado para a função Dixon-Price pelo teste de Wilcoxon pareado e a implementação FGA para a função Rosenbrock e Levy.
- no segundo teste, a implementação FGA apresentou melhores indivíduos para as funções Griewank e Rastrigin. E a implementação FGA+RS destacou-se a função Dixon-Price. Uma observação feita para o segundo teste, que as funções não resultaram diferenças estatisticamente, logo não foi aplicado o teste de Wilcoxon pareado.

Com estes resultados, pode-se concluir que para as funções com muitos ótimos locais, ambas implementações encontraram um desempenho desejável às funções de avaliações Griewank, Rastrigin e Schwefel. Para as funções com um número menor de mínimos locais ou com vales, o modelo FGA+RS gerou melhores valores para a função de Dixon-Price e a FGA se destacou para a função Rosenbrock.

Por esta investigação, pode-se afirmar que nenhuma das implementações possui melhor desempenho sobre a outra. De fato, um melhor resultado, depende de qual função de avaliação será utilizada.

Para trabalhos futuros, recomenda-se seguintes ideias:

- utilizar outras funções de avaliação, como por exemplo, funções *benchmark* de maximização;
- aplicar outros operadores de *crossover* às implementações, como *flat*, discreto;
- executar as implementações para outros parâmetros de testes, afim de observar outras ocorrências;
- comparar as implementações elaboradas utilizando problemas multi-objetivos.
- implementar em outros AGs;

O trabalho proposto foi de fundamental importância no que diz respeito à aprendizagem de fundamentos da área de Inteligência Artificial, Otimização e Estatística.

# Referências

- ARRUDA, E.; KAGAN, N. Estimaco harmnica utilizando algoritmos evolutivos. *X Smpsio de Especialistas em Planejamento da Operao e Expanso Eltrica-X SEPOPE, Florianpolis*, 2006. Citado na pgina 11.
- CAZANGI, R. R. *Operadores de Mutaco e Recombinaco*. Tese (Engenharia Eltrica e de Computaco) — Universidade Estadual de Campinas, 2002. Citado na pgina 18.
- COELHO, L. d. S. et al. Identificaco e controle de processos multivariveis via metodologias avanadas e inteligncia computacional. Florianpolis, SC, 2000. Citado na pgina 20.
- EIBEN, A. E.; SMITH, J. E. *Introduction to evolutionary computing*. [S.l.]: Springer, 2003. v. 53. Citado na pgina 21.
- ESTATCAMP. *Estatstica Bsica: Grficos - boxplot*. 2005. Acesso em: 29 dez. 2016. Disponvel em: <<http://www.portalection.com.br/estatistica-basica/31-boxplot>>. Citado na pgina 29.
- ESTATCAMP. *Estatstica Bsica: Estatsticas descritivas - quartis*. 2005. Acesso em: 29 dez. 2016. Disponvel em: <<http://www.portalection.com.br/estatistica-basica/23-quartis>>. Citado na pgina 29.
- ESTATCAMP. *Tcnicas no paramtricas: Teste de wilcoxon pareado*. 2005. Acesso em: 29 dez. 2016. Disponvel em: <<http://www.portalection.com.br/tecnicas-nao-parametricas/teste-de-wilcoxon-pareado>>. Citado 2 vezes nas pginas 26 e 27.
- FERREIRA, D. V. M. *Aplicaco de algoritmos genticos na elaboraco de horrios institucionais: o estudo de caso do CEFET-MG campus Timteo*. 61 p. Monografia (Graduaco em Engenharia de Computaco) — Centro Federal de Educaco Tecnolgica de Minas Gerais, Timteo, 2015. Citado na pgina 11.
- GOLDBARG, M. C.; GOLDBARG, E. F. G.; NETO, F. D. d. M. Algoritmos evolucionrios na determinaco da configuraco de custo mnimo de sistemas de co-geraco de energia com base no gs natural. *Pesquisa Operacional*, SciELO Brasil, v. 25, n. 2, p. 231–259, 2005. Disponvel em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0101-74382005000200005&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-74382005000200005&nrm=iso)>. Citado na pgina 11.
- GOLDBERG, D. E.; HOLLAND, J. H. Genetic algorithms and machine learning. *Machine learning*, Springer, v. 3, n. 2, p. 95–99, 1988. Citado na pgina 20.
- GONALVES, A. R. *Algoritmos Genticos*. 2013. Disponvel em: <<http://www-users.cs.umn.edu/~andre/arquivos/pdfs/geneticos.pdf>>. Citado na pgina 19.
- GONALVES, P. P. Comparaco entre dois modelos de algoritmos genticos utilizando diferentes funoes de avaliacao de minimizaco. *Centro Federal de Educaco Tecnolgica de Minas Gerais - Campus Timteo*, jul 2016. Citado 5 vezes nas pginas 13, 22, 28, 31 e 33.

HOLLAND, J. H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. [S.l.]: U Michigan Press, 1975. Citado 3 vezes nas páginas 11, 14 e 16.

LINDEN, R. *Algoritmos Genéticos*. [S.l.]: Ciência Moderna, 2012. v. 3. Citado 10 vezes nas páginas 12, 14, 15, 16, 17, 20, 21, 31, 32 e 35.

MICHALEWICZ, Z.; SCHOENAUER, M. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary computation*, MIT Press, v. 4, n. 1, p. 1–32, 1996. Citado na página 20.

NALDI, M. C. *Agrupamento híbrido de dados utilizando algoritmos genéticos*. Tese (Ciências de Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação, 2006. Citado na página 14.

OBITKO, M. Introduction to genetic algorithms. URL <http://www.obitko.com/tutorials/genetic-algorithms>, 1998. Citado na página 17.

R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2016. Acesso em: 29 dez. 2016. Disponível em: <<https://www.R-project.org/>>. Citado na página 28.

RECHENBERG, I. *Cybernetic solution path of an experimental problem*. Franborough, UK: Royal Aircraft Establishment, 1965. Citado na página 21.

RECHENBERG, I. Evolution strategy: Optimization of technical systems by means of biological evolution. *Fromman-Holzboog, Stuttgart*, v. 104, 1973. Citado 5 vezes nas páginas 12, 16, 21, 31 e 35.

ROCHA, G. C. C. *Formação de células de manufatura com abordagem baseada em uma algoritmo genético*. 62 p. Monografia (Graduação em Engenharia de Computação) — Centro Federal de Educação Tecnológica de Minas Gerais, Timóteo, 2015. Citado na página 11.

ROSA, T. d. O.; LUZ, H. S. Conceitos básicos de algoritmos genéticos: Teoria e prática. *XI Encontro de Estudantes de Informática do Tocantins*, p. 27–37, 2009. Citado na página 17.

SCHWEFEL, H.-P. *Evolutionsstrategie und numerische optimierung*. Berlin, Alemanha: Technical University of Berlin, 1975. Citado na página 21.

SCHWEFEL, H.-P. *Numerical optimization of computer models*. [S.l.]: John Wiley & Sons, Inc., 1981. Citado na página 22.

SILVA, R. C. P. Um estudo sobre a autoadaptação de parâmetros na evolução diferencial. Ouro Preto, MG, 2010. Citado na página 11.

SURJANOVIC, S.; BINGHAM, D. *Virtual Library of Simulation Experiments: Test Functions and Datasets: Optimization Test Problems*. 2015. Acesso em: 02 maio 2016. Disponível em: <<http://www.sfu.ca/~ssurjano/optimization.html>>. Citado 6 vezes nas páginas 22, 23, 24, 25, 26 e 31.

TAN, K. C.; YEN, G. G. *EEE World Congress on Computational Intelligence*. 2016. Disponível em: <<http://www.wcci2016.org/>>. Citado na página 11.

WILCOXON, F. Individual comparisons by ranking methods. *Biometrics bulletin*, JSTOR, v. 1, n. 6, p. 80–83, 1945. Citado na página 26.

WILCOXON, F.; KATTI, S.; WILCOX, R. A. Critical values and probability levels for the wilcoxon rank sum test and the wilcoxon signed rank test. *Selected tables in mathematical statistics*, Markham Publishing Co. Chicago, v. 1, p. 171–259, 1970. Acesso em: 29 dez. 2016. Disponível em: <<https://goo.gl/Hd6R0Ne>>. Citado 2 vezes nas páginas 26 e 27.

# Apêndices

# APÊNDICE A – Tabelas do primeiro teste

Neste seção se encontra os dados amostrais do melhor indivíduo de cada implementação do primeiro teste. Os dados foram organizados por função de avaliação. Ressaltando que para este teste a população é formada por 100 indivíduos de 10 genes (dimensão).

Tabela 10 – Dados amostrais do melhor indivíduo aplicado na função Griewank do primeiro teste

Função Griewank		
Menor valor do <i>fitness</i> da população		
Iteração	FGA	FGA+RS
1	2,697E+00	5,882E-01
2	8,100E-01	9,303E-01
3	4,227E-01	8,661E-01
4	1,520E+00	1,080E+00
5	9,730E-01	1,118E+00
6	5,326E-01	7,626E-01
7	1,647E+00	1,497E+00
8	1,686E+00	6,470E-01
9	4,459E-01	2,391E-01
10	4,558E-01	8,041E-01
11	7,227E-01	8,229E-01
12	2,589E-01	6,084E-01
13	2,218E+00	3,500E-01
14	1,351E+00	1,440E+00
15	2,463E-01	8,650E-01
16	4,757E-01	2,917E+00
17	5,516E-01	1,053E+00
18	4,583E-01	6,506E-01
19	5,343E-01	2,096E+00
20	9,386E-01	8,011E-01
21	2,072E+00	3,694E-01
22	1,080E+00	1,983E+00
23	5,422E-01	1,260E-01
24	4,378E-01	7,513E-01
25	1,278E+00	9,659E-02
Menor valor:	1,852E-01	2,666E-01
Média:	1,089E+00	1,006E+00
Maior valor:	2,721E+00	2,592E+00
Desvio padrão	6,707E-01	4,927E-01

Fonte: o autor

Tabela 11 – Dados amostrais do melhor indivíduo aplicado na função Rastrigin do primeiro teste

Função Rastrigin		
Menor valor do <i>fitness</i> da população		
Iteração	FGA	FGA+RS
1	1,375E+00	2,003E+00
2	1,597E+00	5,893E-03
3	2,211E+00	2,600E+00
4	3,092E+00	9,092E-03
5	1,447E+00	2,007E+00
6	9,970E-01	1,364E-02
7	9,958E-01	1,028E-02
8	9,973E-01	2,221E+00
9	9,960E-01	9,997E-01
10	9,974E-01	2,155E+00
11	9,985E-01	3,035E+00
12	1,996E+00	3,065E+00
13	1,593E+00	3,054E+00
14	1,034E-02	5,497E-03
15	5,834E-03	7,488E-03
16	9,966E-01	8,530E-03
17	9,954E-01	4,800E+00
18	2,000E+00	9,995E-01
19	8,590E-03	8,627E-03
20	2,500E+00	2,164E+00
21	3,079E+00	6,701E-03
22	1,000E+00	2,000E+00
23	3,032E+00	4,048E+00
24	1,874E+00	2,007E+00
25	3,089E+00	9,957E-01
Menor valor:	5,633E-03	5,497E-03
Média:	1,906E+00	1,529E+00
Maior valor:	4,082E+00	4,800E+00
Desvio padrão	1,269E+00	1,428E+00

Fonte: o autor

Tabela 12 – Dados amostrais do melhor indivíduo aplicado na função Schwefel do primeiro teste

Função Schwefel		
Menor valor do <i>fitness</i> da população		
Iteração	FGA	FGA+RS
1	3,879E+03	3,699E+03
2	3,952E+03	3,855E+03
3	3,750E+03	3,763E+03
4	3,928E+03	3,699E+03
5	3,774E+03	3,695E+03
6	3,848E+03	3,733E+03
7	3,828E+03	3,770E+03
8	3,843E+03	3,783E+03
9	3,645E+03	3,614E+03
10	3,532E+03	3,886E+03
11	3,733E+03	3,149E+03
12	3,776E+03	3,691E+03
13	3,831E+03	3,570E+03
14	3,653E+03	3,936E+03
15	3,809E+03	3,817E+03
16	3,730E+03	3,842E+03
17	3,830E+03	3,640E+03
18	3,885E+03	3,575E+03
19	3,803E+03	3,837E+03
20	3,708E+03	3,853E+03
21	3,752E+03	3,725E+03
22	3,620E+03	3,693E+03
23	3,758E+03	3,797E+03
24	3,879E+03	3,610E+03
25	3,702E+03	3,731E+03
Menor valor:	3,498E+03	3,149E+03
Média:	3,744E+03	3,719E+03
Maior valor:	3,915E+03	3,936E+03
Desvio padrão	9,901E+01	1,535E+02

Fonte: o autor

Tabela 13 – Dados amostrais do melhor indivíduo aplicado na função Dixon-Price do primeiro teste

Função Dixon-Price		
Menor valor do <i>fitness</i> da população		
Iteração	FGA	FGA+RS
1	5,004E-01	5,001E-01
2	5,006E-01	5,005E-01
3	5,008E-01	5,007E-01
4	5,004E-01	5,007E-01
5	5,005E-01	5,003E-01
6	5,002E-01	5,004E-01
7	5,003E-01	5,007E-01
8	5,008E-01	5,010E-01
9	5,010E-01	5,008E-01
10	5,001E-01	5,004E-01
11	5,006E-01	5,005E-01
12	5,006E-01	5,015E-01
13	5,007E-01	5,004E-01
14	5,009E-01	5,001E-01
15	5,006E-01	5,003E-01
16	5,007E-01	4,455E-01
17	5,019E-01	5,003E-01
18	5,007E-01	5,004E-01
19	5,006E-01	5,006E-01
20	5,013E-01	5,010E-01
21	5,009E-01	5,002E-01
22	5,026E-01	5,003E-01
23	5,036E-01	5,002E-01
24	5,008E-01	5,002E-01
25	5,012E-01	5,011E-01
Menor valor:	5,001E-01	4,455E-01
Média:	5,008E-01	4,983E-01
Maior valor:	5,026E-01	5,015E-01
Desvio padrão	7,572E-04	1,100E-02

Fonte: o autor

Tabela 14 – Dados amostrais do melhor indivíduo aplicado na função Rosenbrock do primeiro teste

Função Rosenbrock		
Menor valor do <i>fitness</i> da população		
Iteração	FGA	FGA+RS
1	1,154E+00	7,258E-01
2	6,659E-01	1,071E+00
3	1,625E+00	5,606E-01
4	9,575E-01	1,787E+00
5	4,869E-01	1,086E+00
6	8,787E+00	3,770E+00
7	5,109E+00	1,422E+00
8	1,053E+00	1,498E+00
9	5,565E-01	4,108E+01
10	1,534E+00	2,507E-01
11	7,757E-01	1,361E+00
12	1,625E+00	5,005E-01
13	3,417E+00	1,261E+00
14	1,415E+00	2,175E-01
15	7,114E-01	1,021E+02
16	9,864E-01	1,338E+00
17	1,488E+00	1,197E+00
18	3,080E-01	8,438E-01
19	7,955E+00	2,004E-01
20	4,367E-01	8,634E-01
21	1,340E+00	1,136E+00
22	9,382E-01	8,579E-01
23	7,129E-01	1,376E+00
24	4,478E-01	4,209E+00
25	1,806E+00	1,149E+00
Menor valor:	3,823E-01	3,055E-01
Média:	1,128E+00	3,149E+00
Maior valor:	5,390E+00	5,186E+01
Desvio padrão	9,874E-01	1,019E+01

Fonte: o autor

Tabela 15 – Dados amostrais do melhor indivíduo aplicado na função Levy do primeiro teste

Função Levy		
Menor valor do <i>fitness</i> da população		
Iteração	FGA	FGA+RS
1	0,000E+00	0,000E+00
2	0,000E+00	0,000E+00
3	0,000E+00	0,000E+00
4	0,000E+00	0,000E+00
5	0,000E+00	0,000E+00
6	0,000E+00	0,000E+00
7	0,000E+00	0,000E+00
8	0,000E+00	0,000E+00
9	0,000E+00	0,000E+00
10	0,000E+00	0,000E+00
11	0,000E+00	0,000E+00
12	0,000E+00	0,000E+00
13	0,000E+00	6,175E-06
14	0,000E+00	0,000E+00
15	0,000E+00	0,000E+00
16	0,000E+00	0,000E+00
17	0,000E+00	0,000E+00
18	0,000E+00	0,000E+00
19	0,000E+00	0,000E+00
20	0,000E+00	0,000E+00
21	0,000E+00	0,000E+00
22	0,000E+00	0,000E+00
23	1,581E-03	0,000E+00
24	7,436E-01	0,000E+00
25	0,000E+00	0,000E+00
Menor valor:	0,000E+00	0,000E+00
Média:	3,799E-21	3,425E-06
Maior valor:	9,497E-20	8,562E-05
Desvio padrão	1,899E-20	1,712E-05

Fonte: o autor

## APÊNDICE B – Tabelas do segundo teste

Neste seção se encontra os dados amostrais do melhor indivíduo de cada implementação do segundo teste. Os dados foram organizados por função de avaliação. Ressaltando que para este teste a população é formada por 200 indivíduos de 20 genes (dimensão).

Tabela 16 – Dados amostrais do melhor indivíduo aplicado na função Griewank do segundo teste

Função Griewank		
Menor valor do <i>fitness</i> da população		
Iteração	FGA	FGA+RS
1	1,760E+00	1,408E+00
2	2,622E+00	2,281E+00
3	1,825E+00	1,484E+00
4	1,236E+00	2,963E+00
5	2,080E+00	2,626E+00
6	2,135E+00	2,149E+00
7	3,497E+00	2,414E+00
8	1,966E+00	1,855E+00
9	2,103E+00	1,799E+00
10	1,032E+00	2,828E+00
11	2,013E+00	1,840E+00
12	1,715E+00	1,884E+00
13	2,090E+00	2,381E+00
14	2,225E+00	1,615E+00
15	2,502E+00	2,868E+00
16	2,294E+00	1,420E+00
17	1,957E+00	2,587E+00
18	1,734E+00	2,636E+00
19	1,402E+00	2,309E+00
20	1,989E+00	1,700E+00
21	1,329E+00	1,852E+00
22	2,049E+00	1,507E+00
23	1,880E+00	2,576E+00
24	1,744E+00	2,000E+00
25	2,210E+00	1,197E+00
Menor valor:	1,032E+00	1,197E+00
Média:	1,976E+00	2,087E+00
Maior valor:	3,497E+00	2,963E+00
Desvio padrão	4,891E-01	5,150E-01

Fonte: o autor

Tabela 17 – Dados amostrais do melhor indivíduo aplicado na função Rastrigin do segundo teste

Função Rastrigin		
Menor valor do <i>fitness</i> da população		
Iteração	FGA	FGA+RS
1	1,003E+00	2,004E+00
2	2,002E+00	2,638E+00
3	6,827E-02	1,672E+00
4	1,011E+00	9,033E-02
5	2,001E+00	1,005E+00
6	2,705E-01	3,642E-02
7	7,850E-02	1,010E+00
8	6,468E-02	7,884E-02
9	1,003E+00	5,119E-02
10	7,111E-02	2,019E+00
11	7,635E-02	7,379E-02
12	5,366E-01	7,911E-02
13	1,998E+00	9,260E-02
14	3,403E-01	1,009E+00
15	3,005E+00	1,014E+00
16	1,009E+00	1,003E+00
17	6,219E-02	5,511E+00
18	3,021E+00	2,008E+00
19	2,003E+00	1,997E+00
20	1,053E+00	1,005E+00
21	7,857E-02	1,125E+00
22	2,003E+00	1,013E-01
23	2,453E-01	1,002E+00
24	1,998E+00	9,984E-01
25	1,010E+00	1,995E+00
Menor valor:	6,219E-02	3,642E-02
Média:	1,041E+00	1,185E+00
Maior valor:	3,021E+00	5,511E+00
Desvio padrão	9,545E-01	1,200E+00

Fonte: o autor

Tabela 18 – Dados amostrais do melhor indivíduo aplicado na função Schwefel do segundo teste

Função Schwefel		
Menor valor do <i>fitness</i> da população		
Iteração	FGA	FGA+RS
1	7,939E+03	7,857E+03
2	7,991E+03	8,048E+03
3	7,905E+03	7,983E+03
4	8,060E+03	7,958E+03
5	8,167E+03	8,111E+03
6	7,897E+03	7,929E+03
7	7,888E+03	7,896E+03
8	8,008E+03	7,795E+03
9	7,832E+03	8,035E+03
10	7,912E+03	7,910E+03
11	8,019E+03	8,006E+03
12	7,872E+03	7,978E+03
13	8,115E+03	7,800E+03
14	8,004E+03	8,011E+03
15	7,897E+03	7,975E+03
16	7,898E+03	8,290E+03
17	8,196E+03	7,961E+03
18	8,204E+03	7,893E+03
19	7,902E+03	7,957E+03
20	8,079E+03	7,994E+03
21	7,919E+03	7,929E+03
22	7,949E+03	7,913E+03
23	7,667E+03	7,886E+03
24	7,952E+03	7,851E+03
25	7,802E+03	7,964E+03
Menor valor:	7,667E+03	7,795E+03
Média:	7,963E+03	7,957E+03
Maior valor:	8,204E+03	8,290E+03
Desvio padrão	1,255E+02	1,015E+02

Fonte: o autor

Tabela 19 – Dados amostrais do melhor indivíduo aplicado na função Dixon-Price do segundo teste

Função Dixon-Price		
Menor valor do <i>fitness</i> da população		
Iteração	FGA	FGA+RS
1	5,391E-01	5,014E-01
2	5,122E-01	5,036E-01
3	5,012E-01	5,024E-01
4	5,066E-01	5,152E-01
5	5,077E-01	5,028E-01
6	5,033E-01	5,008E-01
7	5,014E-01	5,028E-01
8	5,012E-01	5,083E-01
9	5,020E-01	5,027E-01
10	5,025E-01	5,053E-01
11	5,012E-01	5,014E-01
12	5,019E-01	5,077E-01
13	5,059E-01	5,059E-01
14	5,021E-01	5,058E-01
15	5,140E-01	5,013E-01
16	5,052E-01	5,025E-01
17	5,013E-01	5,006E-01
18	5,023E-01	5,007E-01
19	5,009E-01	5,025E-01
20	5,189E-01	5,015E-01
21	5,018E-01	5,011E-01
22	5,017E-01	5,011E-01
23	5,077E-01	5,024E-01
24	5,037E-01	5,012E-01
25	5,028E-01	5,113E-01
Menor valor:	5,009E-01	5,006E-01
Média:	5,060E-01	5,037E-01
Maior valor:	5,391E-01	5,152E-01
Desvio padrão	8,276E-03	3,625E-03

Fonte: o autor

Tabela 20 – Dados amostrais do melhor indivíduo aplicado na função Rosenbrock do segundo teste

Função Rosenbrock		
Menor valor do <i>fitness</i> da população		
Iteração	FGA	FGA+RS
1	9,043E+00	8,221E+00
2	7,376E+00	5,912E+00
3	9,179E+00	7,873E+00
4	7,468E+00	1,051E+01
5	4,488E+00	6,600E+00
6	4,024E+00	8,882E+00
7	7,719E+00	7,469E+00
8	9,894E+00	1,030E+01
9	8,588E+00	8,079E+00
10	8,143E+00	7,097E+00
11	1,311E+01	5,560E+00
12	6,243E+00	7,522E+00
13	9,899E+00	5,483E+00
14	5,340E+00	5,351E+00
15	9,491E+00	1,108E+01
16	6,642E+00	9,039E+00
17	5,845E+00	1,174E+01
18	1,031E+01	8,620E+00
19	5,630E+00	1,278E+01
20	6,612E+00	7,385E+00
21	1,082E+01	9,337E+00
22	8,941E+00	7,596E+00
23	9,540E+00	9,559E+00
24	8,421E+00	8,257E+00
25	8,145E+00	9,746E+00
Menor valor:	4,024E+00	5,351E+00
Média:	8,036E+00	8,400E+00
Maior valor:	1,311E+01	1,278E+01
Desvio padrão	2,123E+00	1,948E+00

Fonte: o autor

Tabela 21 – Dados amostrais do melhor indivíduo aplicado na função Levy do segundo teste

Função Levy		
Menor valor do <i>fitness</i> da população		
Iteração	FGA	FGA+RS
1	0,000E+00	0,000E+00
2	0,000E+00	0,000E+00
3	0,000E+00	0,000E+00
4	0,000E+00	0,000E+00
5	0,000E+00	0,000E+00
6	0,000E+00	0,000E+00
7	0,000E+00	0,000E+00
8	0,000E+00	0,000E+00
9	0,000E+00	0,000E+00
10	0,000E+00	0,000E+00
11	0,000E+00	0,000E+00
12	0,000E+00	0,000E+00
13	0,000E+00	0,000E+00
14	0,000E+00	0,000E+00
15	0,000E+00	0,000E+00
16	0,000E+00	0,000E+00
17	0,000E+00	0,000E+00
18	0,000E+00	0,000E+00
19	0,000E+00	0,000E+00
20	0,000E+00	0,000E+00
21	0,000E+00	0,000E+00
22	0,000E+00	0,000E+00
23	0,000E+00	0,000E+00
24	0,000E+00	0,000E+00
25	0,000E+00	0,000E+00
Menor valor:	0,000E+00	0,000E+00
Média:	0,000E+00	0,000E+00
Maior valor:	0,000E+00	0,000E+00
Desvio padrão	0,000E+00	0,000E+00

Fonte: o autor