

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ENGENHARIA ELÉTRICA
BACHARELADO EM ENGENHARIA ELÉTRICA**

MARCO ANTONIO ITABORAHY FILHO

**ANÁLISE DE ALGORITMOS GENÉTICOS E EVOLUÇÃO
DIFERENCIAL PARA OTIMIZAÇÃO DE FUNÇÕES NÃO-LINEARES
MULTIMODAIS**

TRABALHO DE CONCLUSÃO DE CURSO

**PONTA GROSSA
2018**

MARCO ANTONIO ITABORAHY FILHO

**ANÁLISE DE ALGORITMOS GENÉTICOS E EVOLUÇÃO
DIFERENCIAL PARA OTIMIZAÇÃO DE FUNÇÕES NÃO-LINEARES
MULTIMODAIS**

Trabalho de Conclusão de Curso apresentado(a) como requisito parcial à obtenção do título de Bacharel em Engenharia Elétrica, do Departamento Acadêmico de Engenharia Elétrica, da Universidade Tecnológica Federal do Paraná.

Orientador(a): Prof. Dr. Hugo Valadares Siqueira

Coorientador: Prof. Dr. Mauricio dos Santos Kaster

PONTA GROSSA

2018



TERMO DE APROVAÇÃO

ANÁLISE DE ALGORITMOS GENÉTICOS E EVOLUÇÃO DIFERENCIAL PARA OTIMIZAÇÃO DE FUNÇÕES NÃO-LINEARES MULTIMODAIS

por

MARCO ANTONIO ITABORAHY FILHO

Este Trabalho de Conclusão de Curso foi apresentado em 30 de Novembro de 2018 como requisito parcial para a obtenção do título de Bacharel em Engenharia Elétrica. O(A) candidato(a) foi arguido(a) pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Hugo Valadares Siqueira
Orientador(a)

Prof. Dr. Flávio Trojan
Membro Titular

Prof^ª. Dr^ª. Marcella Scoczynski Ribeiro
Martins
Membro Titular

Prof. Dr. Josmar Ivanqui
Responsável pelos TCC

Prof. Msc. Jeferson José Gomes
Coordenador do Curso

Dedico este trabalho a minha família que
sempre esteve comigo.

AGRADECIMENTOS

Aos meus pais, pelo amor incondicional, apoio constante e pela paciência.

Às minhas irmãs por terem me suportado por todos estes anos.

À Beatriz pelos melhores abraços do mundo.

Aos Kibes, minha família escolhida a dedo, pela amizade mais sincera que alguém pode pedir.

Ao Bingo, meus melhores amigos que eu nunca conheci, pelas horas de discussão.

Ao Professor Hugo, por toda dedicação e empenho neste último ano.

E a todos os meus colegas do LICON pelo companheirismo e pelo café.

Muito além, nos confins inexplorados da região mais brega da Borda Ocidental desta Galáxia, há um pequeno sol amarelo e esquecido... Girando em torno deste sol, a uma distância de cerca de 148 milhões de quilômetros, há uma planetinha verde-azulado absolutamente insignificante, cujas formas de vida, descendentes de primatas, são tão extraordinariamente primitivas que ainda acham que relógios digitais são uma grande ideia. (ADAMS, Douglas, 1979).

RESUMO

ITABORAHY FILHO, Marco Antonio. **Análise de Algoritmos Genéticos e Evolução Diferencial para Otimização de Funções Não-Lineares Multimodais**. 2018. 58 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2018.

Nossa sociedade vem procurando resolver problemas cada vez mais complexos à medida que a tecnologia avança. Neste contexto, métodos de solução tradicionais podem, muitas vezes, não gerar respostas com a qualidade ou velocidade necessárias. Por essa razão, foram investigados neste trabalho os Algoritmos Genéticos e a Evolução Diferencial, que são categorizados como Algoritmos Evolutivos, já que tratam de métodos de otimização baseados na evolução das espécies Darwiniana. Foram encontrados na literatura diferentes formas de implementação que mostram aprimoramentos dos métodos de otimização tradicional. Foram descritas 23 diferentes estratégias, as quais foram avaliadas utilizando três diferentes funções *Benchmark*. As respostas foram comparadas observando a velocidade de convergência e a qualidade da saída obtida.

Palavras-chave: Algoritmos Evolutivos. Algoritmos Genéticos. Evolução Diferencial. Otimização. Funções Benchmark.

ABSTRACT

ITABORAHY FILHO, Marco Antonio. **Analyses of Genetic Algorithms and Differential Evolution for Non-Linear Multimodal Functions Optimization**. 2018. 58 p. Final Coursework (Bachelor's Degree in Electric Engineering) – Federal University of Technology – Paraná. Ponta Grossa, 2018.

Our society has been looking to solve more complex problems every day, with this context in mind, traditional solving methods can, many times, fall short on generating answers with the required speed or quality . For this reason, we studied, on this paper, Genetic Algorithms and Differential Evolution algorithms, both are classified as Evolutionary Algorithms, optimization methods based on the evolution of the species by Natural Selection. Were described 23 different Evolutionary Algorithms strategies and tested them using three different Benchmark functions, the answers were them studied and compared, by their speed of convergence and the quality of their outputs. There were found methods that, not only are an improvement over traditional optimization methods, but also over Evolutionary Algorithms that are currently more known and used.

Keywords: Evolutionary Algorithms. Genetic Algorithms. Differential Evolution. Optimization. Benchmark Functions.

LISTA DE ALGORITMOS

Algoritmo 1 – Exemplo de um Algoritmo Genético Canônico	25
Algoritmo 2 – Exemplo de um Algoritmo de Evolução Diferencial	33

LISTA DE ILUSTRAÇÕES

Figura 1 – Ilustração das Variações de Bicos encontrados nos Tentilhões de Galápagos.	17
Figura 2 – Estrutura do DNA.	18
Figura 3 – Exemplo de Crossover em um Cromossomo.	19
Figura 4 – Fluxograma do Funcionamento Básico de um AG.	24
Figura 5 – Exemplo Simples de uma Roleta.	26
Figura 6 – Exemplo Simples de um Torneio.	27
Figura 7 – Exemplo de um <i>Crossover</i> de ponto	28
Figura 8 – Exemplo de Crossover de Ponto em um Problema Binário.	29
Figura 9 – Fluxograma do Funcionamento Básico de um ED.	33
Figura 10 – Demonstração de uma Mutação em ED.	34
Figura 11 – Exemplo de <i>Crossover</i> Binário em ED.	36
Figura 12 – Exemplo de <i>Crossover</i> Exponencial em ED	36
Figura 13 – Função Exponencial.	39
Figura 14 – Função Exponencial.	39
Figura 15 – Função Exponencial.	40
Figura 16 – Função Exponencial.	40
Figura 17 – Função Rastrigin.	41
Figura 18 – Função Rastrigin.	42
Figura 19 – Função Rastrigin.	42
Figura 20 – Função Rastrigin.	43
Figura 21 – Função Schwefel.	44
Figura 22 – Função Schwefel.	44
Figura 23 – Função Schwefel.	45
Figura 24 – Função Schwefel.	45
Figura 25 – Boxplot do Fitness da Função Exponencial.	48
Figura 26 – Boxplot do Fitness da Função Exponencial.	50
Figura 27 – Boxplot do Fitness da Função Schwefel.	52

LISTA DE TABELAS

Tabela 1 – Resultados das Simulações da Função Exponencial	47
Tabela 2 – Resultados das Simulações da Função Rastrigin	49
Tabela 3 – Resultados das Simulações da Função Schwefel	51

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	14
1.2	ORGANIZAÇÃO DO TRABALHO	14
2	INSPIRAÇÃO BIOLÓGICA	16
2.1	HISTÓRIA DA TEORIA DA EVOLUÇÃO	16
2.2	CONCEITOS BÁSICOS DA EVOLUÇÃO	17
2.3	HISTÓRICO DOS ALGORITMOS EVOLUTIVOS	21
2.4	APLICAÇÃO DOS CONCEITOS DE EVOLUÇÃO EM OTIMIZAÇÃO	21
3	ALGORITMOS GENÉTICOS	23
3.1	CONCEITOS BÁSICOS	23
3.2	SELEÇÃO	24
3.2.1	Roleta	25
3.2.2	Torneio	26
3.3	CROSSOVER	27
3.4	MUTAÇÃO	29
3.5	OUTRAS ESTRATÉGIAS	30
3.6	VARIAÇÕES DE ALGORITMOS GENÉTICOS UTILIZADAS	30
4	EVOLUÇÃO DIFERENCIAL	32
4.1	CONCEITOS BÁSICOS	32
4.2	PSEUDOCÓDIGO DE UM ED GENÉRICO	33
4.3	MUTAÇÃO	34
4.4	CROSSOVER	35
4.5	SELEÇÃO	36
4.6	VARIAÇÕES DE EVOLUÇÃO DIFERENCIAL	36
5	FUNÇÕES BENCHMARK	38
5.1	FUNÇÃO EXPONENCIAL	38
5.2	FUNÇÃO RASTRIGIN	41
5.3	FUNÇÃO SCHWEFEL	42
6	RESULTADOS E DISCUSSÃO	46
6.1	FUNÇÃO EXPONENCIAL	46
6.2	FUNÇÃO RASTRIGIN	49
6.3	FUNÇÃO SCHWEFEL	51
6.4	DISCUSSÃO DOS RESULTADOS	53
7	CONCLUSÃO	55
	REFERÊNCIAS	56

1 INTRODUÇÃO

Algoritmos Evolutivos (AE) são metaheurísticas de otimização e busca eficientes e robustos, os quais foram desenvolvidos inspirados na teoria da evolução das espécies de Darwin.

Com o avanço da tecnologia, a humanidade vem enfrentando tarefas cada vez mais complexas e de maior dimensionalidade. Dessa maneira, novos métodos para solução de problemas devem ser desenvolvidos, já que, muitas vezes, estas não poderiam ser encontradas utilizando algoritmos convencionais, como métodos estocásticos ou métodos de gradiente.

Um exemplo disto é a solução para o Problema Booleano dos Trios Pitagóricos apresentado (HEULE; KULLMANN; MAREK, 2016), um famoso problema sugerido em 1980 por Ronald Graham cuja prova matemática foi feita inteiramente computacionalmente, gerando mais de 200 Terabytes de dados. Isto é considerado impossível de ser lido na íntegra por um ser humano.

O constante avanço da velocidade de computação responde a Lei de Moore, a qual enuncia que a cada dois anos a densidade de transistores em um circuito integrado dobra (MOORE, 2006). Assim, faz-se necessário o uso de métodos computacionais cada vez mais avançados para a solução de problemas de alta complexidade hoje encontrados.

Para muitos problemas reais, métodos exatos são pouco eficientes, podendo levar dias para encontrar soluções pouco atraentes do ponto de vista ótimo. O mesmo se aplica para tarefas de otimização em que algoritmos como Otimização Estocástica e Otimização por Gradiente conseguem atingir o objetivo em um período de tempo razoável, principalmente quando uma solução apenas praticável não é o suficiente e a precisão de resposta ao problema é crítica (VOß, 2001). Uma alternativa é a utilização de métodos de otimização bio-inspirados, como Algoritmos Evolutivos. Este trabalho estuda dois métodos de otimização desta categoria: Algoritmos Genéticos (AG) e Evolução Diferencial (ED).

1.1 OBJETIVOS

O principal objetivo deste trabalho é realizar um estudo sobre Algoritmos Evolutivos para otimização, fazendo uma análise comparativa entre diferentes estratégias utilizando funções *benchmark* para testes de qualidade e velocidade de convergência. Como diferentes tipos de problemas podem necessitar diferentes métodos de solução, espera-se que esta análise possa ser utilizada no futuro para que a escolha de um algoritmo mais eficiente, dependendo das características conhecidas do problema.

Para isso, foram escolhidos para este trabalho 13 variações de Algoritmos Genéticos e 10 variações de Evolução Diferencial aplicados em 3 funções tridimensionais, cada uma apresentando diferentes características. Isto permite que uma análise de comportamento de cada algoritmo em diferentes situações. A análise da resposta de cada um destes às diferentes funções permite traçar paralelos com problemas de otimização reais como, por exemplo, encontrar valores ótimos de ganhos para controladores adaptativos não-lineares, problemas de ordenamento ou problemas de divisão de zonas eleitorais, para que seja possível identificar quais métodos de otimização são os mais indicados para obtermos uma resposta precisa com máxima eficiência.

Como objetivos específicos podemos citar:

- Aprofundar a discussão em torno dos algoritmos citados e suas versões;
- Implementar computacionalmente as 23 versões de AE, validando-as;
- Exibir os resultados computacionais para cada cenário, fazendo uma análise crítica sobre as vantagens e desvantagens de cada abordagem.

1.2 ORGANIZAÇÃO DO TRABALHO

No segundo capítulo deste trabalho iremos fazer uma breve revisão dos conceitos da evolução das espécies, que é a inspiração biológica por trás dos Algoritmos Evolutivos e como este conceito é aplicado em otimização. No Terceiro Capítulo serão abordados o funcionamento e as variações de Algoritmos Genéticos.

No quarto capítulo serão exploradas as características da Evolução Diferencial e as diferentes estratégias DE implementação. Para o Capítulo 5 serão apresentadas as três diferentes Funções *Benchmark* utilizadas para testes dos AE.

No Capítulo 6 são apresentados os resultados encontrados das simulações

realizadas para teste e feita uma breve discussão destes resultados para, então, no último Capítulo ser realizada a conclusão dos trabalhos.

2 INSPIRAÇÃO BIOLÓGICA

Algoritmos Evolutivos, como o próprio nome sugere, são inspirados nos princípios de evolução por seleção natural, descritos pela primeira vez por Charles Darwin em 1859 em seu famoso livro "A Origem das Espécies"(DARWIN, 1959). Neste capítulo iremos fazer uma breve discussão dos processos envolvidos na Evolução, focando nos processos naturais mais relevantes para nossa área.

2.1 HISTÓRIA DA TEORIA DA EVOLUÇÃO

Em todo o mundo foram elaboradas diversas ideias procurando responder a uma das maiores perguntas da humanidade: A origem da vida e diferenças entre espécies. A primeira grande teoria científica da evolução foi proposta por Jean-Baptiste Lamarck. Sua teoria sugeria, em parte, que indivíduos se adaptavam ao ambiente de acordo com suas necessidades e que órgãos e membros sofriam alterações de acordo com o “uso ou desuso” (LAMARCK, 1976).

Por exemplo, como girafas utilizavam seus pescoços para alcançar folhas de árvores muito altas, os mesmos cresciam através das gerações. Elefantes utilizavam suas trombas para muitas funções, logo elas ficariam maiores e mais complexas com o tempo (LAMARCK, 1976).

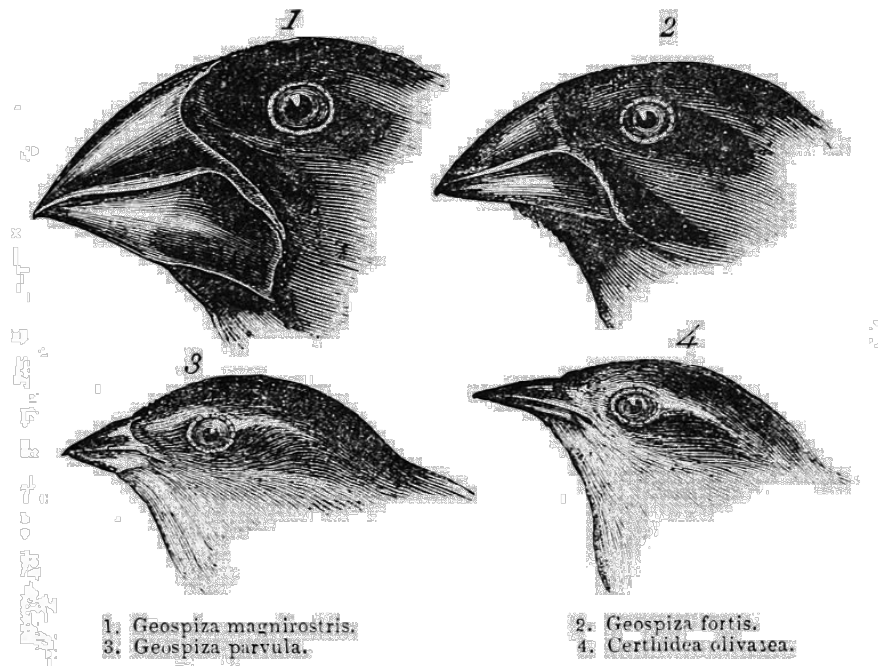
Em 1858 os naturalistas Charles Darwin e Alfred Russel Wallace publicaram suas descobertas, desenvolvidas em paralelo. Ambos concluíram que os indivíduos não se adaptam ao ambiente, mas que, em realidade, aqueles mais bem adaptados tem uma maior chance de sobreviver. Indivíduos com características biológicas que os permitissem uma maior facilidade em escapar de predadores, encontrar comida e parceiros sexuais conseguiria então passar estas características para seus filhos (DARWIN, 1959).

Um exemplo dos efeitos da evolução encontrado por Darwin em suas viagens pelo arquipélago de Galápagos, são os tentilhões de Galápagos. Estas espécies de pássaros descendem de um mesmo ancestral, porém cada espécie acabou isolada em uma ilha diferente do arquipélago, com pequenas diferenças em seus ambientes. A diferença ambiental mais óbvia encontrada foram os tipos de alimento disponíveis

às aves.

O que se observou foi que as diferentes espécies de tentilhões possuíam bicos de formatos e tamanhos diferentes (Figura 1), especializados em alcançar, processar e comer as frutas encontradas em duas respectivas ilhas.

Figura 1 – Ilustração das Variações de Bicos encontrados nos Tentilhões de Galápagos.



Fonte: "Darwin's finches," por John Gould (Domínio Público).

À época do desenvolvimento desta teoria, não se sabia ainda qual era o processo natural que permitia que características de uma geração pudesse passar para a próxima. Este problema só foi começar a ser resolvido em 1866 com o Australiano Gregor Johann Mendel e seus experimentos com hereditariedade em ervilhas (MENDEL; MANGELSDORF, 1965). Entretanto, suas ideias só foram utilizadas para o estudo da evolução a partir de 1925, após uma série de investigações lideradas por Thomas Hunt Morgan. A partir de então foi possível criar uma linha de pesquisa baseada em genética populacional (MUKHERJEE, 2016).

2.2 CONCEITOS BÁSICOS DA EVOLUÇÃO

De acordo com o estudo da genética, cada ser vivo possui um código genético singular composto por uma fita helicoidal de chamado de DNA (Ácido Desoxirribonucleico). O DNA (Figura 2) é um composto orgânico que pode ser interpretado como

sendo uma série de instruções para cada molécula de um ser vivo. Quando “lidas” por organelas especializadas, estas informações indicam quais aminoácidos ou proteínas devem ser produzidos. Este processo controla todas as funções básicas do corpo, desde a absorção de água em uma célula a até a produção de novas células.

Nos seres humanos o DNA encontra-se no núcleo celular, separado em 23 pares de cromossomos, para um total de 46 cromossomos. Cada um possui inúmeros genes, e é a partir da sequência genética que as características de um indivíduo são guardadas, desde como o funcionamento de cada célula do corpo deve agir até características físicas do indivíduo.

Figura 2 – Estrutura do DNA.



Fonte: Public Domain Pictures.

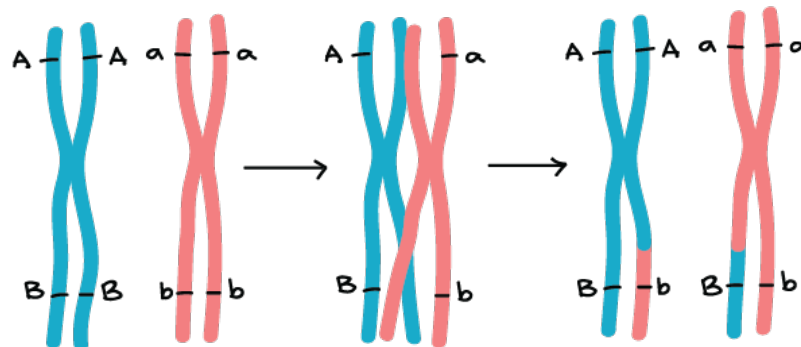
A constituição genética de cada ser vivo é chamada de genótipo, o qual guarda todas as informações de como cada indivíduo do nível molecular ao nível macro. Este material é passado dos pais para a prole. Dessa forma, características de uma geração são passadas para a próxima através do cruzamento utilizando gametas (GRIFFITHS et al., 2008).

As últimas são as células reprodutoras dos seres vivos produzidos pelo processo de meiose: uma divisão celular na qual cada célula produzida recebe um cromossomo de cada par. Nos seres humanos cada gameta possui 23 cromossomos (GRIFFITHS et al., 2008).

Um dos processos relevantes ao nosso trabalho e que ocorre durante a reprodução é o *crossover*. Durante ele acontece uma troca de materiais genéticos entre os pares de cromossomos resultando em duas novas sequências de genes com partes

de ambos os cromossomos originais, como exemplificado na Figura 3.

Figura 3 – Exemplo de Crossover em um Cromossomo.



Fonte: Khan Academy.

O *crossover* é importante para que filhos possuam uma mistura de características provenientes dos 4 cromossomos provenientes dos pais (um par do pai e um par da mãe), sendo que cada um é originário dos avós. Esta mistura gera uma maior variabilidade genética, o que pode, muitas vezes ser vantajosa para uma espécie.

O conjunto de características físicas e biológicas de cada indivíduo resultantes da genética, como nos seres humanos - altura, cor da pele, etc. é conhecido como fenótipo (MENDEL; MANGELSDORF, 1965). O fenótipo de um indivíduo é determinado por seus genes, ou seja, por seus pais e qualquer mutação que possa ter ocorrido ao genótipo de um indivíduo.

No cruzamento, os gametas masculinos e femininos se unem, gerando o embrião de um novo indivíduo. Este possui cada metade proveniente dos gametas que o geraram, com genótipos de ambos os pais e formando, assim, uma nova sequência genética única (GRIFFITHS et al., 2008).

Este embrião então se multiplica rapidamente, gerando um novo indivíduo, que após a maturação do indivíduo, irá repetir o processo de cruzamento, passando suas características adquiridas por seus pais ou mutação para a próxima geração, e assim por diante.

Outro processo genético relevante ao nosso estudo é o conceito de mutação. Durante a duplicação de cromossomos, processo fundamental para a divisão celular, pode ocorrer, aleatoriamente, erros na construção desses novos cromossomos, gerando mudanças espúrias no genótipo daquela célula. Caso ocorra uma mutação em um gameta, pode ser criado um novo indivíduo com partes de seu genótipo sendo

originais (GRIFFITHS et al., 2008).

A mutação é possivelmente o principal processo para a ocorrência da evolução. Apesar do fato da maioria das mutações ocorridas serem insignificantes, estas pequenas mudanças podem causar importantes alterações no fenótipo que, em alguns casos são benéficas ao indivíduo. Estas variações podem, ao longo do tempo se espalhar para toda uma população (MUKHERJEE, 2016).

Ao contrário do que Lamarck acreditava, a evolução não influencia diretamente no fenótipo de um indivíduo. De fato, trata-se de um processo não determinístico que age nos indivíduos e, conseqüentemente, acaba influenciando uma população ou espécie como um todo.

Um outro processo natural responsável pela evolução é conhecido como Seleção Natural. Esta atua na diferença das taxas de sobrevivência entre indivíduos de acordo com as variações entre seus fenótipos. A variabilidade entre indivíduos decorrem da mutação e podem levar a diferenças em como eles interagem com o ambiente. Ademais, outros indivíduos podem melhorar ou piorar as chances de sobreviver até a vida adulta, quando estes podem então se reproduzir (MUKHERJEE, 2016).

Durante a luta pela sobrevivência, para a maioria das espécies, a taxa de mortalidade é altíssima, fazendo com que os animais que conseguem alcançar a maturidade e se reproduzir sejam, em geral, os indivíduos mais fortes e adaptados aos perigos enfrentados.

Por exemplo, na savana, gazelas devem ser capazes de escapar de predadores, como os leões. Ao longo das gerações, os genes que geravam indivíduos fortes e ágeis criam outros com maiores chances de sobreviver e passar adiante este fenótipo. No caso dos tentilhões de Darwin, as mutações que faziam com que a alimentação das aves fosse mais eficiente se espalharam, fazendo que animais de diferentes ilhas tivessem bicos especializados para as frutas presentes.

A seleção natural ocorre graças à luta pela sobrevivência de indivíduos. Comida, segurança contra outros animais ou o ambiente, capacidade de atrair parceiros, etc., são todas lutas diárias que os seres vivos devem suceder constantemente para que seus genes possam ser passados para futuras gerações. De acordo com o biólogo evolucionário Richard Dawkins (DAWKINS; FLORSHEIM, 2001) este é o maior objetivo que cada indivíduo busca alcançar. Isto pode explicar as inúmeras espécies que morrem no processo de gerar ou proteger a nova geração, como os polvos que

morrem de fome para manter seus ovos em segurança ou abelhas, cujos os machos morrem após o cruzamento com a rainha.

2.3 HISTÓRICO DOS ALGORITMOS EVOLUTIVOS

Os primeiros trabalhos publicados sobre computação evolutiva surgiram nos meados dos anos 1950. Em 1958 e 1959 Friedberg (BACK; FOGEL; MICHALEWICZ, 1997) apresenta dois artigos sobre o que conhecemos hoje como *machine learning*. Porém, nesta época estes eram mais utilizados para criação de programas de cálculos de expressões matemáticas (BACK; FOGEL; MICHALEWICZ, 1997).

Os primeiros algoritmos genéticos para otimização foram concebidos por John Henry Holland em 1967, o conceito foi então melhor desenvolvido por Holland e seu estudante David Goldberg durante a década de 1970 (HOLLAND, 1975). Já a primeira proposta de Evolução Diferencial ocorreu em 1997 por Reiner Storn e Kenneth Price (STORN; PRICE, 1997), a qual foi proposta como uma solução mais simples quando comparada ao AG.

2.4 APLICAÇÃO DOS CONCEITOS DE EVOLUÇÃO EM OTIMIZAÇÃO

Ambas as estratégias citadas na seção anterior utilizam o conceito de evolução para alcançar soluções ótimas de otimização: Soluções candidatas para o problema abordado são equivalentes a um indivíduo (ou cromossomo) em um grupo de animais de mesma espécie. Cada um possui um genótipo e um fenótipo. O primeiro, neste caso, são coordenadas no espaço de resposta do problema e podem ser representados por um vetor. Já o fenótipo destes indivíduos é conhecido como *fitness*, ou o valor da avaliação do mesmo na função objetivo que se deseja otimizar (MICHALEWICZ, 1996).

O *fitness* de cada indivíduo é calculado utilizando uma função determinada anteriormente, que depende das características do problema. Por exemplo, em um problema de maximização, o valor de *fitness* pode ser o próprio valor de saída. Já para minimização muitas vezes é utilizado o inverso da saída da função. O *fitness* indica quão bem adaptada aquela solução é e quanto maior seu valor, melhor a resposta (GOLDBERG, 1989).

Nos algoritmos evolutivos, os candidatos estão sujeitos a operadores equivalentes aos processos naturais de evolução, de modo que estes mesmos processos levam à criação de espécies especializadas de animais com vistas à sobrevivência em certos ambientes e criem resposta de otimização (GOLDBERG, 1989).

Algoritmos Evolucionários são inicializados gerando aleatoriamente um número de indivíduos no espaço de busca do problema. Tal quantidade deve ser condizente com a complexidade da busca já que um número muito alto pode causar lentidão no processo, devido ao número excessivo de simulações realizadas por geração, e um número muito baixo pode necessitar de muitas gerações para a convergência (MICHALEWICZ, 1996).

Os operadores genéticos supracitados são o *crossover* e a mutação e, por fim seleção. Indivíduos com altos valores de *fitness* tem maiores chances de sobreviver ao processo de seleção e, conseqüentemente, passar seus genes para as próximas gerações. isto é repetido em toda geração até um critério de parada ser alcançado (MITCHELL, 1996).

O desejo de manter indivíduos de *fitness* alto é conhecido como elitismo, o qual garante que o melhor indivíduo já encontrado pelo algoritmo não seja eliminado nas próximas gerações (MICHALEWICZ, 1996).

É importante ressaltar que o desenvolvimento de Algoritmos Evolutivos foi um processo em que muitas das características ideais para otimização foram descobertas a partir de processos empíricos.

3 ALGORITMOS GENÉTICOS

Os Algoritmos Genéticos foram desenvolvidos por Holland e Goldberg e surgiram como um método de busca e otimização. A inspiração biológica advém das mecânicas da seleção natural e genética, combinando os conceitos de sobrevivência do mais forte e utilizando *strings* com troca de informação aleatória utilizando regras de transição probabilística (GOLDBERG, 1989). Neste capítulo iremos discutir o funcionamento destas estratégias de otimização.

3.1 CONCEITOS BÁSICOS

Algoritmos genéticos funcionam com base em regras simples: a primeira geração é criada aleatoriamente, no espaço de solução do problema. A partir dela, as próximas gerações são criadas através de regras probabilísticas envolvendo a geração anterior. Este processo nada mais é do que a cópia de *strings* completas ou de frações delas. As versões mais simples se resumem a três operadores: Seleção, *Crossover* e Mutação (MICHALEWICZ, 1996).

A seleção é o processo pelo qual um indivíduo é escolhido baseado em sua aptidão, ou valor de *fitness*, para participar da concepção da nova geração. Isto pode ser feito de diversas maneiras, sendo as mais populares a Roleta e o Torneio (BACK; FOGEL; MICHALEWICZ, 1997).

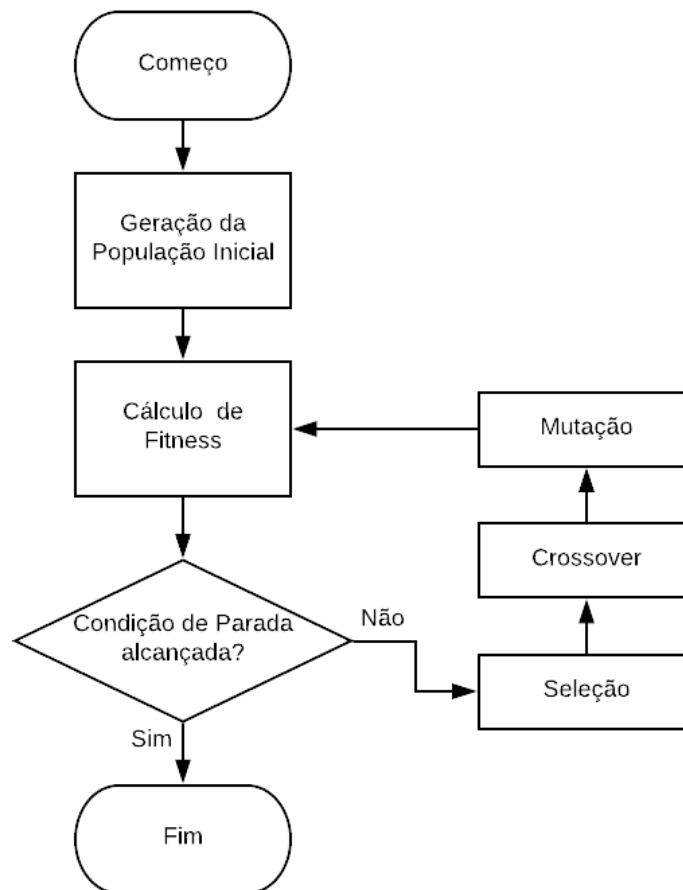
O *crossover* é o operador genético que realiza o equivalente na natureza do cruzamento: indivíduos realizam uma troca genética, gerando, assim, novos indivíduos que herdam características da geração anterior. As diferentes variações que serão aplicadas neste trabalho serão o *Crossovers* de Ponto, Aritmético e SBX (do inglês *Simulated Binary Crossover*).

O último operador, a mutação é considerada por Goldberg um operador secundário. Apesar disso, em muitas situações é impossível alcançar uma resposta adequada sem mutação. Trata-se de uma variação aleatória de um gene de um indivíduo. A sua taxa de ocorrência deve ser baixa para que possa ser mantido um nível adequado de elitismo (MICHALEWICZ, 1996).

O funcionamento de um AG genérico pode ser resumido como uma sucessão

de operadores genéticos executados em cada indivíduo de uma geração para a criação da próxima. Um fluxograma simplificado de todo o processo pode ser visto na Figura 4.

Figura 4 – Fluxograma do Funcionamento Básico de um AG.



Fonte: Autoria Própria.

Além disso, o pseudocódigo apresentado no Algoritmo 1 é baseado na proposta canônica descrita por Michalewicz em 1996 (MICHALEWICZ, 1996).

3.2 SELEÇÃO

A seleção dos pais para cruzamento, em algoritmos genéticos, é feita utilizando, principalmente, dois diferentes métodos, os quais serão discutidos e utilizados neste trabalho: a Roleta e o Torneio.

Algoritmo 1 – Exemplo de um Algoritmo Genético Canônico

- 1: **início**
 - 2: $t = 0$
 - 3: inicie a população $P(t)$
 - 4: avalie $P(t)$
 - 5: **enquanto** Condição de parada não é alcançada **faça**
 - 6: $t = t + 1$
 - 7: selecione $P(t)$ através de $P(t - 1)$
 - 8: *crossover* de $P(t)$
 - 9: mutação de $P(t)$
 - 10: avalie $P(t)$
 - 11: **finaliza enquanto**
 - 12: **fim**
-

Fonte: Autoria própria.

3.2.1 Roleta

O método da roleta, ou seleção proporcional por *fitness* é, de acordo com Jinghui Zhong, a forma de seleção mais utilizada. É criada de uma roleta virtual viciada na qual os pais são então escolhidos. Cada indivíduo ocupa uma área da roleta diretamente proporcional a seu *fitness* (BACK; FOGEL; MICHALEWICZ, 1997).

Durante sua execução são calculados os valores de *fitness* de cada indivíduo. É feita, então, a soma de todos os *fitness* para a obtenção do tamanho da roleta que será utilizada, ou seja, o número de posições que podem ser escolhidas na seleção. Este número é dividido pelo *fitness* de cada indivíduo para determinar quantas posições serão ocupadas por cada um (GOLDBERG, 1989).

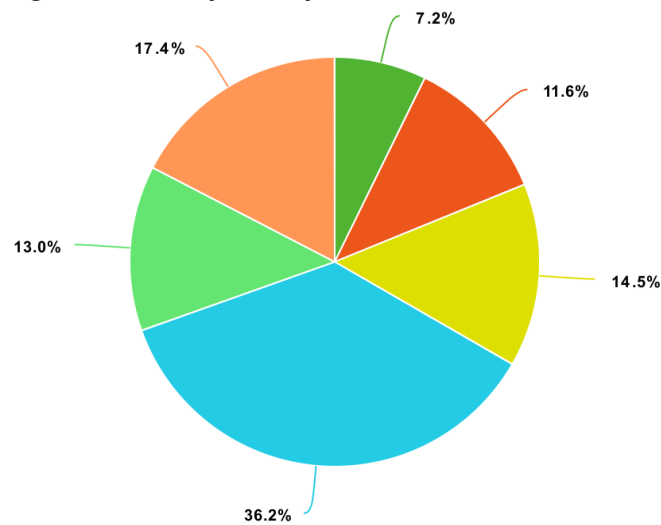
A chance de um indivíduo ser escolhido pode ser calculada através da Equação (1), na qual i é a posição de um indivíduo na matriz, N é o número total de indivíduos, $PROB(i)$ é a chance de um indivíduo ser selecionado e $Fit(i)$ é o valor calculado do *fitness*.

$$PROB(i) = \frac{fit(i)}{\sum_{i=1}^N fit(n)} \quad (1)$$

A Figura 5 mostra um esquema de como é feito o processo.

A seleção é feita utilizando um ponteiro que, após girar, irá apontar para uma área da roleta correspondente a um indivíduo. Por razões claras, indivíduos com valores de *fitness* melhores terão mais chances de ser escolhidos para participar do cruzamento. Usualmente, o número seleções feitas é igual ao número de indivíduos existentes na população e um indivíduo pode acabar sendo selecionado múltiplas ve-

Figura 5 – Exemplo Simples de uma Roleta.



Fonte: Autoria Própria.

zes. Isso, entretanto, traz a tona um dos problemas esperados mais frequentes do método: possibilidade dos indivíduos de melhor valor de *fitness* acabem não sendo selecionados, o que diminuiria a eficácia do algoritmo.

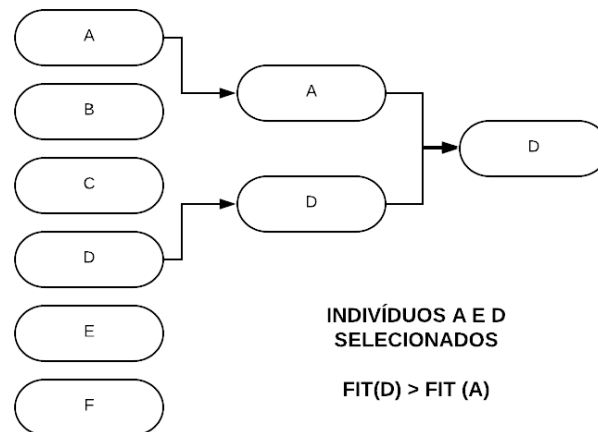
Uma variação da roleta é chamada de *Stochastic Universal Sampling*. Neste, indivíduos são selecionados com espaçamentos iguais entre eles dentro da roleta. Este método aumenta a chance de indivíduos de baixo *fitness* serem selecionados, aumentando assim a variabilidade genética da população.

3.2.2 Torneio

A seleção por torneio é uma estratégia de seleção baseada no *fitness* dos indivíduos, sem considerar o *fitness* total da população. A ideia básica é comparar múltiplos agentes (normalmente dois) escolhidos aleatoriamente e então fazer a seleção, escolhendo o o indivíduo de melhor *fitness* para participar da criação da próxima geração, como pode ser observado na Figura 6. Este é considerado, na maioria dos casos, um método de seleção melhor quando comparado ao método da roleta, já que a pressão seletiva exercida é menor (ZHONG et al., 2005).

As duas variações de torneio que serão utilizadas neste trabalho são: o torneio simples, em que cada indivíduo pode participar inúmeras vezes e o Torneio da Morte, no qual os perdedores são então eliminados. Para o torneio que não mata, ocorre uma pressão seletiva menor já indivíduos de melhor *fitness* possuem uma chance

Figura 6 – Exemplo Simples de um Torneio.



Fonte: Autoria Própria.

menor de serem eliminados da população. Na segunda estratégia, toda a população é garantida de passar pelo torneio, aumentando, assim, a diversidade genética da população (ZHONG et al., 2005).

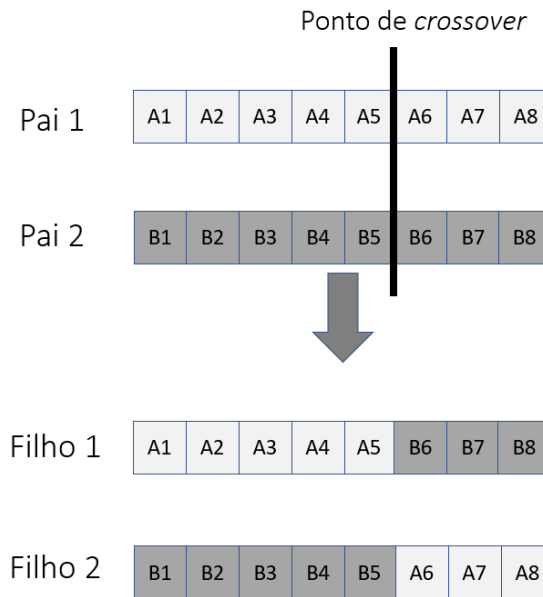
3.3 CROSSOVER

O *crossover* é um operador genético que simula o *crossover* que ocorre entre cromossomos e, nos algoritmos genéticos, equivale ao cruzamento de dois indivíduos na natureza. Aqui são criados novos indivíduos que contêm partes do genótipo dos pais, processo, em geral, é feito em pares, com dois pais e dois filhos. Serão analisados neste trabalho três diferentes vertentes: de Ponto, Aritmético e o *Simulated Binary Crossover* (SBX).

O *Crossover* de Ponto é simples: é escolhido um ponto no gene dos pais onde será feito um corte destes genes, em seguida estes pedaços são recombinados, criando novos indivíduos com parte dos genes de ambos os pais. Um exemplo de um *Crossover* de Ponto pode ser visto na Figura 7. Este tipo é o mais comumente utilizado e é o mais similar ao *crossover* ao processo natural (HOLLAND, 1975).

O *Crossover* Aritmético é feito utilizando duas médias ponderadas, criando filhos no espaço entre ambos os pais. A Equação 2 é a utilizada e mostra que os filhos

Figura 7 – Exemplo de um Crossover de ponto



Fonte: Autoria Própria.

são, de fato, combinações lineares dos pais (GOLDBERG, 1994)

$$\begin{aligned} \text{Filho1} &= \alpha \text{Pai1} + (1 - \alpha) \text{Pai2} \\ \text{Filho2} &= (1 - \alpha) \text{Pai1} + \alpha \text{Pai2} \end{aligned} \quad (2)$$

na qual α é um número aleatório entre 0 e 1 gerado para cada indício de crossover.

Algoritmos genéticos foram, originalmente, desenvolvidos para problemas binários, e, ao longo do tempo foram adaptados para tratar problemas de números reais. Assim, algumas características do *crossover* de ponto foram perdidas. O *crossover* SBX, ou Binário Simulado, foi desenvolvido para recuperar estas características. Como é possível ver na Figura 8, a média dos valores dos filhos em um *crossover* binário é igual o valor médio dos pais (DEB; KUMAR, 1995).

A fórmula para o *crossover* SBX é como na Equação 3:

$$\begin{aligned} \text{Filho1} &= 0,5(\text{Pai1} + \text{Pai2}) - 0,5\beta(\text{Pai1} - \text{Pai2}) \\ \text{Filho2} &= 0,5(\text{Pai1} + \text{Pai2}) + 0,5\beta(\text{Pai2} - \text{Pai1}) \end{aligned} \quad (3)$$

Figura 8 – Exemplo de Crossover de Ponto em um Problema Binário.

PAIS	FILHOS												
<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">1</td> </tr> </table> = 37	1	0	0	1	0	1	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">0</td> </tr> </table> = 34	1	0	0	0	1	0
1	0	0	1	0	1								
1	0	0	0	1	0								
<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">0</td> </tr> </table> = 26	0	1	1	0	1	0	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">1</td> </tr> </table> = 29	0	1	1	1	0	1
0	1	1	0	1	0								
0	1	1	1	0	1								
Média = 31,5	Média = 31,5												

Fonte: Autoria Própria.

sendo que

$$\beta = (2u)^{\frac{1}{M+1}}, \text{ caso } u < 0,5. \quad (4)$$

$$\beta = \frac{0,5}{1-u}^{\frac{1}{M+1}}, \text{ caso } u \geq 0,5$$

sendo u um número aleatório entre 0 e 1 e M uma constante, em geral entre 5 e 200.

Este operador é o que permite o refinamento das soluções.

3.4 MUTAÇÃO

A função da mutação é, de certa forma, adicionar informações novas à genética dos indivíduos, o que pode ser benéfico à sua população. Após o *crossover* cada indivíduo têm uma chance igual de ter um de seus genes mutados para um valor aleatório. A probabilidade de isto acontecer é dada pela Taxa de Mutação. Serão utilizados neste trabalho duas variações de mutação: fixa e dinâmica (HONG; WANG, 1996).

Na mutação fixa, a mais frequente, a taxa não muda durante todo o processo. Na dinâmica, esta varia com o tempo de forma que quando maior o valor do *fitness* médio de toda população, maior será a chance dela ocorrer. Isto é feito para aumentar a variabilidade genética, quando já os indivíduos tendem a estar muito próximo um dos outros. Este operador é o que permite a realização de busca global.

3.5 OUTRAS ESTRATÉGIAS

Duas estratégias empregadas neste trabalho são os conceitos de Busca Local e *Niching*.

Busca local se refere a um método em que um indivíduo procura em sua vizinhança próxima um ponto de melhor *fitness* (ELMIHOUB et al., 2004). O que se faz é com que o melhor indivíduo passe por um processo de refinamento na qualidade da sua solução atual, o que pode gerar respostas mais adequadas. Este processo pode ser utilizado em conjunto com qualquer configuração dos algoritmos genéticos.

Niching é processo de penalizar indivíduos que se aproximarem muito um dos outros. Isto é feito dividindo o *fitness* destes indivíduos (MAHFOUD, 1995). Tal procedimento ajuda a população a manter uma diversidade genética maior, o que pode buscar gerar uma maior diversidade genética na população.

3.6 VARIAÇÕES DE ALGORITMOS GENÉTICOS UTILIZADAS

Neste estudo foram desenvolvidas 13 variações de algoritmos genéticos para comparação de suas capacidades de busca. É entendido que a taxa de *crossover* é de 100% e a taxa de mutação é de 10% quando não citadas. Estas variações são:

- AG 1
Conhecido como o Algoritmo Genético Clássico, utiliza seleção por Roleta, *Crossover* de Ponto com taxa de *crossover* de 70% e mutação fixa;
- AG 2
Seleção por Roleta, *Crossover* de ponto e mutação fixa;
- AG 3
Seleção por Roleta na qual pais e filhos participam da seleção ao mesmo tempo, *Crossover* de Ponto e Mutação Fixa;
- AG 4
Seleção por Torneio Simples, *Crossover* de ponto e mutação fixa;
- AG 5
Seleção por Torneio Simples com pais e filhos participando da seleção ao

mesmo tempo, *Crossover* de Ponto e mutação fixa;

- AG 6
Seleção por Torneio da Morte, *Crossover* de Ponto e Mutação fixa;
- AG 7
Seleção por Roleta com *Stochastic Universal Sampling*, *Crossover* de Ponto, Mutação fixa;
- AG 8
Seleção por Torneio Simples com uso de Niching, *Crossover* de Ponto e Mutação Fixa;
- AG 9
Seleção por Torneio Simples com Busca Local em toda geração, *Crossover* de Ponto e Mutação fixa;
- AG 10
Seleção por Torneio Simples com Busca Local a cada 10 gerações, *Crossover* de Ponto e Mutação fixa;
- AG 11
Seleção por Torneio Simples, *Crossover* Aritmético e Mutação Fixa;
- AG 12
Seleção por Torneio Simples, *Crossover* de Ponto e Mutação Dinâmica;
- AG 13
Seleção por Torneio Simples, *Crossover* SBX e Mutação Fixa.

4 EVOLUÇÃO DIFERENCIAL

Apresentado pela primeira vez em 1995 por Reiner Storn e Kenneth Price, a Evolução Diferencial (ED) foi proposta como uma solução robusta para problemas de otimização contínua (STORN; PRICE, 1997). As soluções são encontradas a partir dos mesmos processos de Seleção, *Crossover* e Mutação que Algoritmos Genéticos, porém aqui estes operadores possuem características diferentes.

Neste capítulo serão abordadas as características e variações da Evolução Diferencial (ED), estratégia de otimização inspiradas no conceito de algoritmos evolucionários. A ED é um método populacional simples, mas ao mesmo tempo poderoso (STORN; PRICE, 1997).

4.1 CONCEITOS BÁSICOS

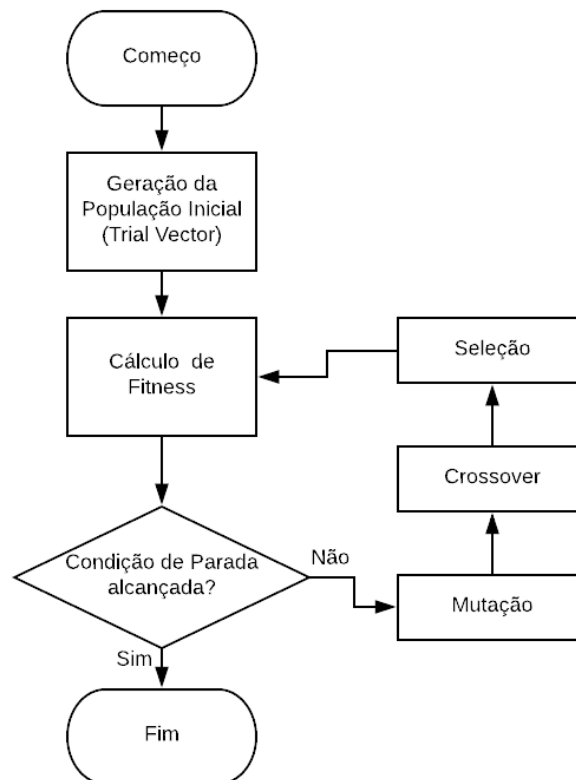
Indivíduos utilizam não só os genes dos pais mas também podem usar diretamente o melhor indivíduo de cada geração em seus processos genéticos de otimização. Os passos envolvidos no desenvolvimento da evolução diferencial são relativamente simples, se comparados aos de um Algoritmo Genético (TUSAR; FILIPIC, 2007).

O funcionamento de um algoritmo de ED segue os seguintes passos: Uma população de vetores (agentes) inicial é criada aleatoriamente, como é usual. O vetor que está sendo trabalhado a cada geração é chamado de *Target Vector*. É criado, então, um segundo agente através da mutação, que é feita adicionando a diferença ponderada entre dois indivíduos. O vetor mutado é chamado de *Donor Vector*.

O último passo, então, por uma combinação com o *Target Vector*, criando assim o chamado *Trial Vector*. Este processo é o equivalente ao *crossover*.

Na seleção, o *Trial Vector* atual é comparado com o *Target Vector* e aquele com o maior *fitness* passa para a próxima geração. Um fluxograma simplificado de um ED pode ser observado a seguir na Figura 9.

Figura 9 – Fluxograma do Funcionamento Básico de um ED.



Fonte: Autoria Própria.

4.2 PSEUDOCÓDIGO DE UM ED GENÉRICO

O pseudocódigo do funcionamento básico da Evolução Diferencial pode ser também visto no Algoritmo 2.

Algoritmo 2 – Exemplo de um Algoritmo de Evolução Diferencial

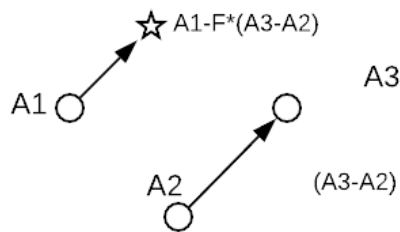
-
- 1: **início**
 - 2: $t = 0$
 - 3: inicie a população $P(t)$
 - 4: avalie $P(t)$
 - 5: **enquanto** Condição de parada não é alcançada **faça**
 - 6: $t = t + 1$
 - 7: crie vetor através da mutação
 - 8: faça *crossover* entre os *Target Vectors* e vetores mutados gerando *Trial Vectors*
 - 9: compare *Target Vector* e *Trial Vector* gerando um novo *Target Vector* $P(t)$
 - 10: avalie $P(t)$
 - 11: **finaliza enquanto**
 - 12: **fim**
-

Fonte: Autoria própria.

4.3 MUTAÇÃO

Para cada *Target Vector* é criado um vetor mutado, ou *Donor Vector*, através do processo de somar um indivíduo aleatório a uma diferença ponderada de um número de outros indivíduos. O que isso significa é que um indivíduo irá dar um passo no espaço de resposta. A direção e tamanho deste passo é determinada pela posição de outros indivíduos e uma variável determinada anteriormente, como mostra a Figura 10.

Figura 10 – Demonstração de uma Mutaç o em ED.



Fonte: Autoria Pr pria.

Ao contr rio do Algoritmo Gen tico, no qual a era considerado um operador gen tico secund rio, na Evolu o Diferencial ela   o principal operador e a maior fonte de diversidade entre diferentes estrat gias.

A Muta o em ED possui dois tipos de varia es: a primeira   o n mero de diferen as ponderadas que ser o utilizadas, podendo ser, normalmente, uma ou duas diferen as. A equa o Equa o 5 demonstra um exemplo de cada um destes casos utilizando indiv duos aleatoriamente escolhidos:

$$v_i = x_{r_1} + F(x_{r_2} - x_{r_3}) \tag{5}$$

$$v_i = x_{r_1} + F((x_{r_2} - x_{r_3}) + (x_{r_4} - x_{r_5}))$$

onde v_i   o indiv duo i mutado, x_{r_1} a x_{r_5} s o indiv duos aleat rios do *Target Vector*.

A pr xima varia o da muta o remete a quais s o os indiv duos que s o utilizados para a muta o. Aqui ser o consideradas tr s varia es: *Rand*, *best* e *Target-to-best*. A primeira muta o utiliza apenas indiv duos aleat rios; j  a muta o tipo *best* aborda o indiv duo com o melhor valor de *fitness* como o indiv duo a ser somado na

mutação, enquanto na *target-to-best* o indivíduo de melhor *fitness* é utilizado na diferença ponderada da mutação. Um exemplo de cada tipo de mutação pode ser visto nas equações Equação 6:

$$\begin{aligned}
 v_i &= x_{r1} + F(x_{r2} - x_{r3}) \text{ Mutação tipo Rand} \\
 v_i &= x_{best} + F(x_{r1} - x_{r2}) \text{ Mutação tipo Best} \\
 v_i &= x_{r1} + F(x_{best} - x_{r2}) \text{ Mutação tipo Target-to-Best}
 \end{aligned} \tag{6}$$

nas quais, x_{r1} a x_{r3} são indivíduos aleatórios do *Target Vector*, x_{best} é o melhor indivíduo da população e F é uma constante real determinada entre 0 e 2 que define o tamanho do passo que será dado. Foi determinante para este trabalho, um valor constante de $F = 0,8$, baseado na literatura encontrada.

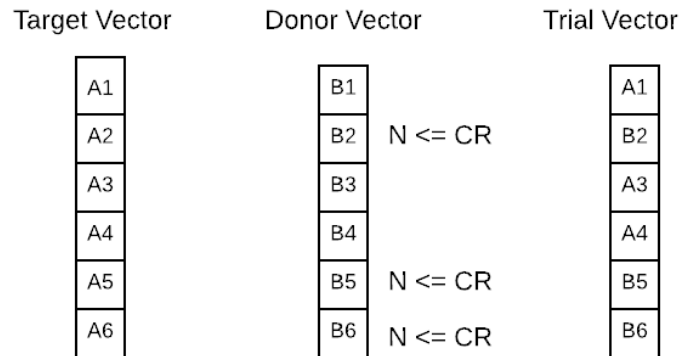
4.4 CROSSOVER

O *crossover* em Evolução Diferencial acontece criando um novo agente chamado de *Trial Vector*, utilizando os *Target Vector* e *Donor Vector*. Serão utilizadas duas variações de *crossover* neste trabalho, o Binário e Exponencial (LIN; QING; FENG, 2011).

O primeiro caso segue os seguintes passos: Para cada posição do *trial vector* no *crossover* binário, é realizado um experimento de Bernoulli cuja chance de sucesso é determinada pela variável CR . Caso ocorra um sucesso, o indivíduo que irá preencher aquela posição do *trial vector* será o *donor vector*. Caso contrário, este será o *target vector*, como pode ser visto na Figura 11. O valor utilizado de CR neste trabalho foi de 20%, baseado na literatura já existente.

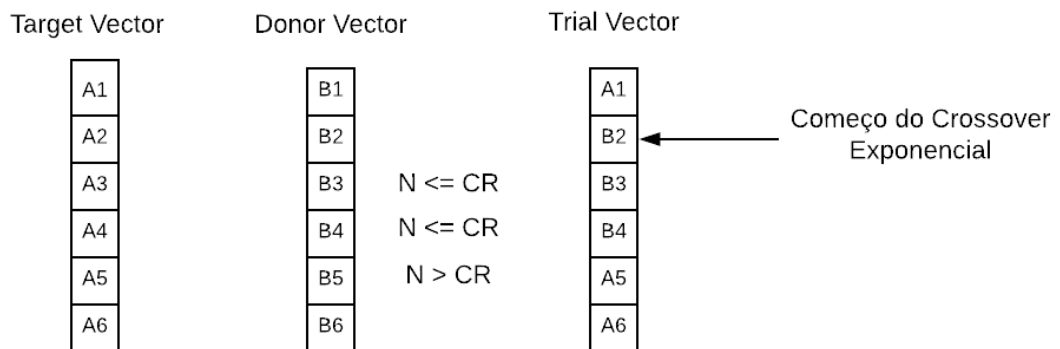
O *crossover* exponencial em ED segue o seguinte processo: É escolhido um ponto aleatório na população atual. A partir deste ponto, indivíduos fazem um *crossover* binário tradicional, porém, assim que houver um fracasso do experimentos de Bernoulli, o processo de *crossover* é encerrado e os espaços do *trial vector* restantes são preenchidos pelos indivíduos respectivos do *target vector*. Isto faz com que a chance do próximo indivíduo ser preenchido pelo *donor vector* caia exponencialmente. Um exemplo de um *crossover* exponencial pode ser visto na Figura 12.

Figura 11 – Exemplo de *Crossover* Binário em ED.



Fonte: Autoria Própria.

Figura 12 – Exemplo de *Crossover* Exponencial em ED



Fonte: Autoria Própria.

4.5 SELEÇÃO

A seleção é o operador mais simples na Evolução Diferencial, já que é realizada de forma gulosa. O atual *target vector* e *trial vector* são comparados e o de maior valor de *fitness* é então selecionado.

4.6 VARIAÇÕES DE EVOLUÇÃO DIFERENCIAL

Algoritmos de Evolução Diferencial tem uma nomenclatura padrão de classificações que facilita a identificação de qual estratégia que está sendo utilizada. esta dá-se da seguinte forma: "Tipo de Mutação / Número de Diferenças Ponderadas na Mutação / Tipo de *Crossover*". Seguindo estas regras, um algoritmo com mutação tipo

Best com duas diferenças ponderadas e *crossover* binário pode ser descrito desta forma: *Best/2/Bin*.

Para este trabalho foram utilizados 10 diferentes variações da Evolução Diferencial, sendo elas:

- ED 1: *Rand/1/Bin*
- ED 2: *Best/1/Bin*
- ED 3: *Rand/2/Bin*
- ED 4: *Target – to – Best/2/Bin*
- ED 5: *Best/1/Bin*
- ED 6: *Rand/1/Exp*
- ED 7: *Best/2/Exp*
- ED 8: *Rand/2/Exp*
- ED 9: *Target – to – Best/2/Exp*
- ED 10: *Best/1/Exp*

No próximo capítulo aplicaremos as variações do AG e da ED em problemas de otimização de funções em 3 dimensões.

5 FUNÇÕES BENCHMARK

Para avaliação do potencial de busca e otimização dos 23 algoritmos descritos nos capítulos anteriores foram utilizadas três diferentes funções *Benchmark* em 3 dimensões do tipo $z = f(x,y)$.

Funções Benchmark são função genéricas onde todas suas características são previamente conhecidas. Sendo assim, é possível utilizar este tipo de função para testar o funcionamento de nossas funções do Otimização sob condições controladas.

As funções escolhidas foram: Uma função Exponencial, uma função simples, com apenas um ponto de mínimo e um ponto de máximo e pode ser análogo a problemas com apenas uma possível solução ótima, como por exemplo o Problema do Caixeiro Viajante. A função Rastrigin e a Função Schwefel, estas duas funções são função multimodais, podendo simular problemas reais, como por exemplo, a otimização de ganhos um controlador adaptativo ou o problema de recorte de distritos eleitorais. As características de cada uma destas funções serão apresentadas a seguir.

5.1 FUNÇÃO EXPONENCIAL

A função exponencial utilizada neste trabalho consiste de uma função simples com apenas um vale e um pico. É descrita pela fórmula da Equação 7:

$$z = xe^{-(x^2+y^2)} \quad (7)$$

limitada em x e y entre -2 e 2 .

Nesta função os nossos algoritmos evolutivos foram projetados para maximização, que é sabido que neste caso o ponto de ótimo esta em $x = \frac{1}{\sqrt{2}}$ ou, aproximadamente $1,414$ E $y = 0$ e $z = \frac{1}{\sqrt{e*2}}$, ou $0,4288$.

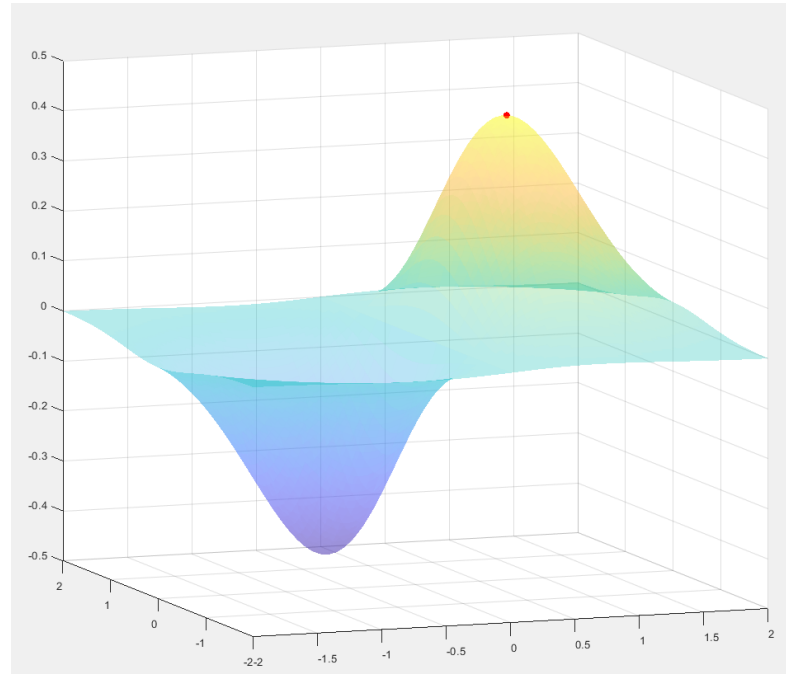
Para o cálculo de *fitness* desta função foi utilizada a Equação 8:

$$fitness = z\sqrt{2e} \quad (8)$$

de forma que no ponto de máximo teremos um valor de *fitness* unitário.

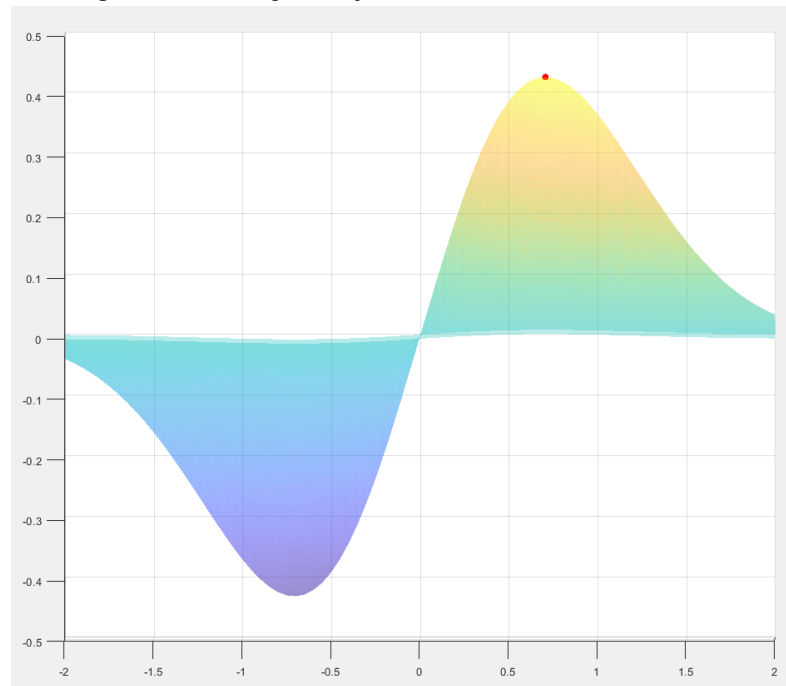
A função exponencial será apresentada e seu ponto de máximo global, então, nas figuras: Figura 13, Figura 14, Figura 15 e Figura 16.

Figura 13 – Função Exponencial.

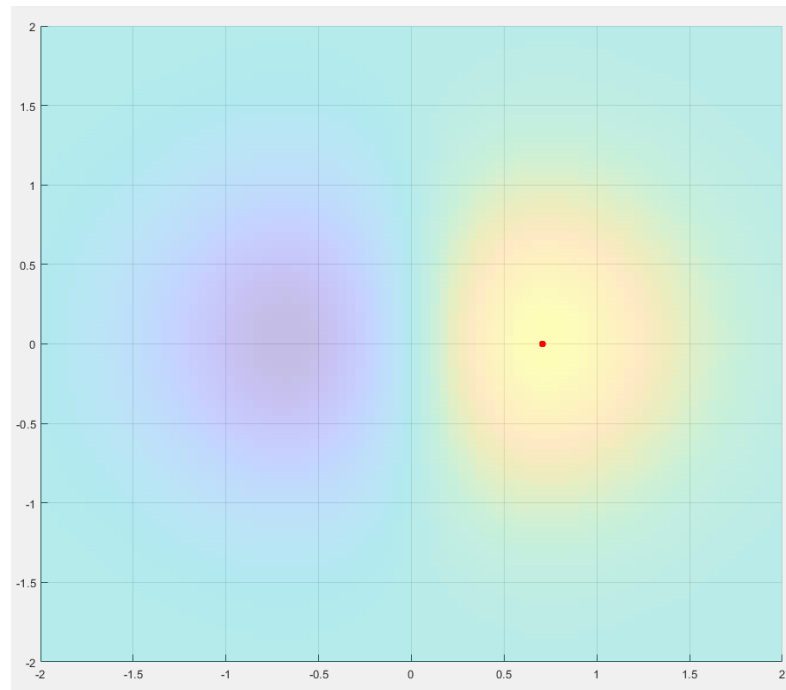


Fonte: Autoria Própria.

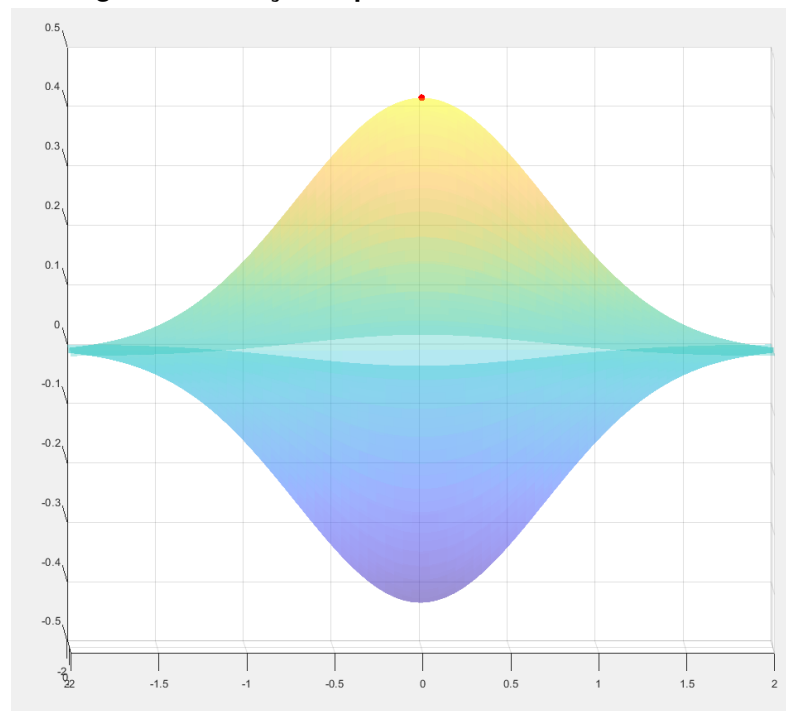
Figura 14 – Função Exponencial.



Fonte: Autoria Própria.

Figura 15 – Função Exponencial.

Fonte: Autoria Própria.

Figura 16 – Função Exponencial.

Fonte: Autoria Própria.

5.2 FUNÇÃO RASTRIGIN

A função Rastrigin foi analisada, neste trabalho, no intervalo de x e y entre -5 e 5 . Neste, a função possui 100 picos e 81 vales, sendo assim, ideal para testes de algoritmos de solução de problemas não-lineares e multimodais. A equação da função Rastrigin em três dimensões é dada por:

$$z = 20 + x^2 + y^2 - 10(\cos(2\pi x) + \cos(2\pi y)) \quad (9)$$

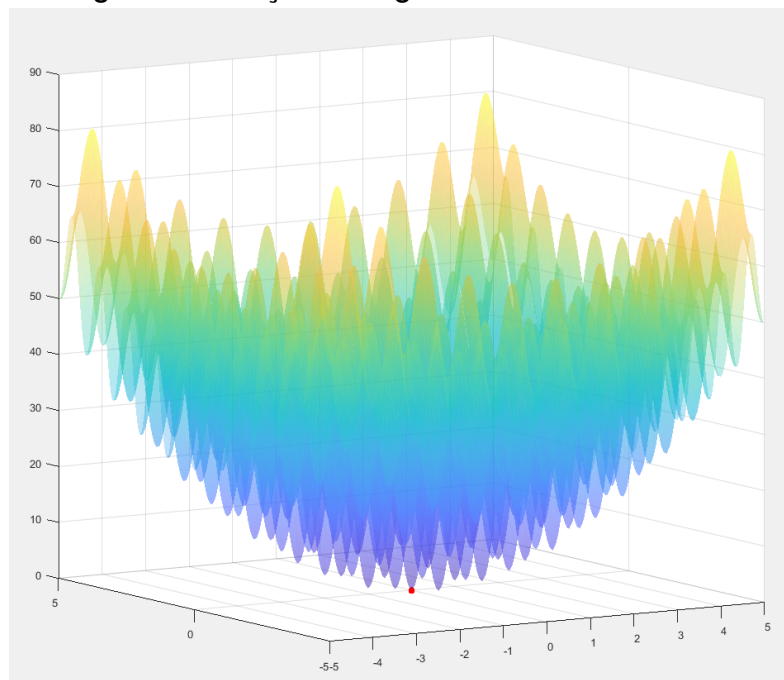
A função Rastrigin é utilizada para minimização. Sabe-se que o ponto de mínimo global fica localizado no ponto $x = 0$, $y = 0$ e $z = 0$. Com isso em mente a fórmula do cálculo de *fitness* da função utilizada é:

$$fitness = \frac{1}{1 + z} \quad (10)$$

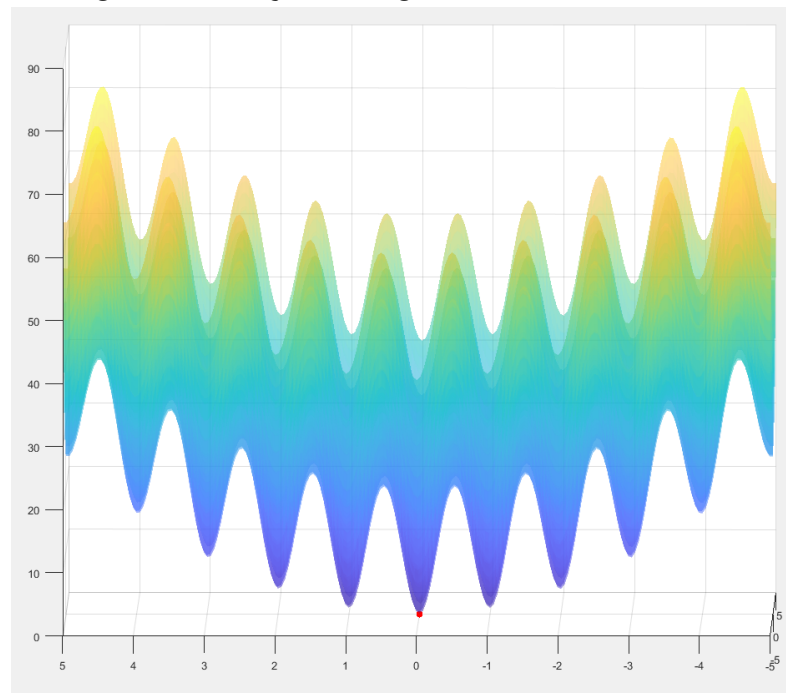
para que seja alcançado um *fitness* máximo unitário.

A Função Rastrigin e seu ponto de mínimo global pode ser observada nas figuras: Figura 17 Figura 18, Figura 19 e Figura 20:

Figura 17 – Função Rastrigin.



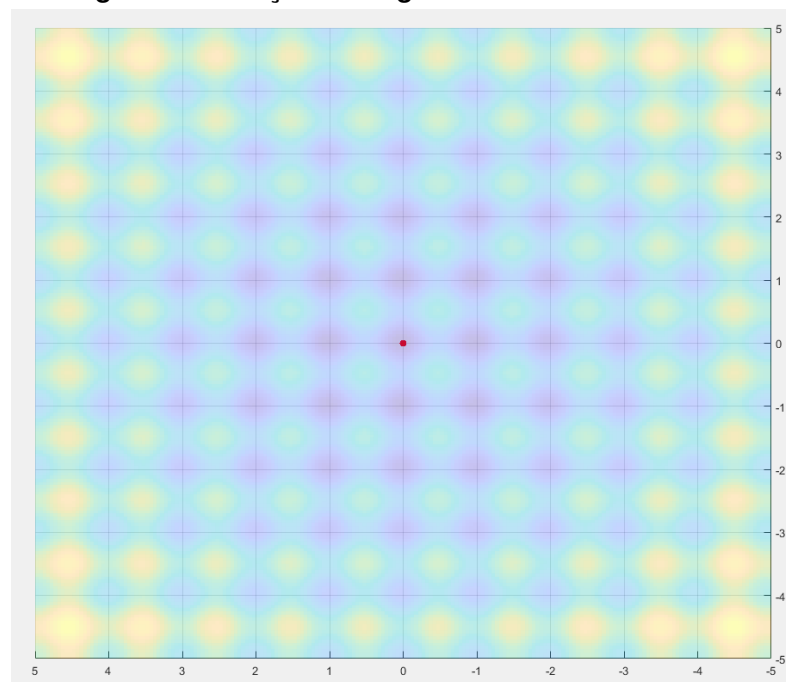
Fonte: Autoria Própria.

Figura 18 – Função Rastrigin.

Fonte: Autoria Própria.

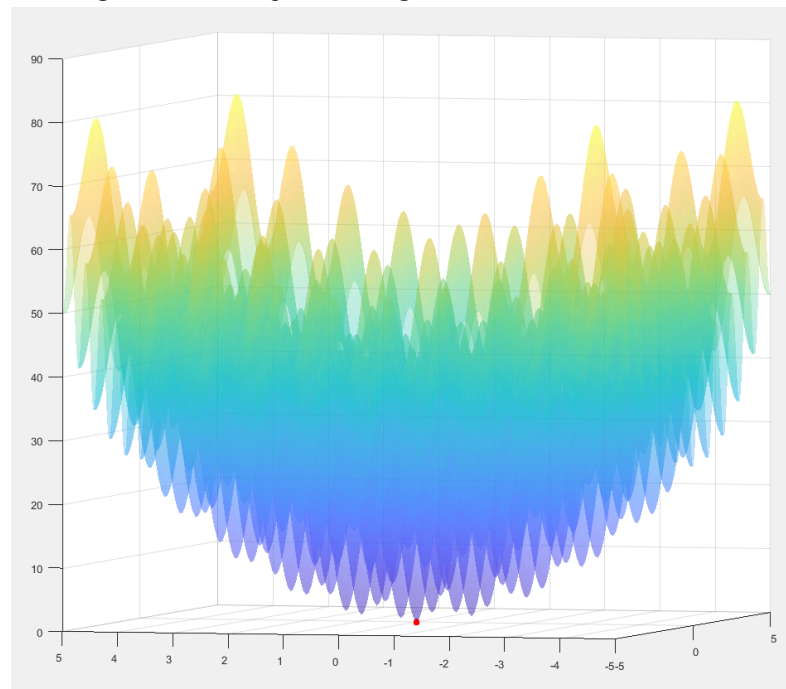
5.3 FUNÇÃO SCHWEFEL

A função Schwefel é multimodal não-separável e apresenta cerca de 60 picos e 50 vales no intervalo em que x e y estão entre -500 e 500 . A versão tridimensional

Figura 19 – Função Rastrigin.

Fonte: Autoria Própria.

Figura 20 – Função Rastrigin.



Fonte: Autoria Própria.

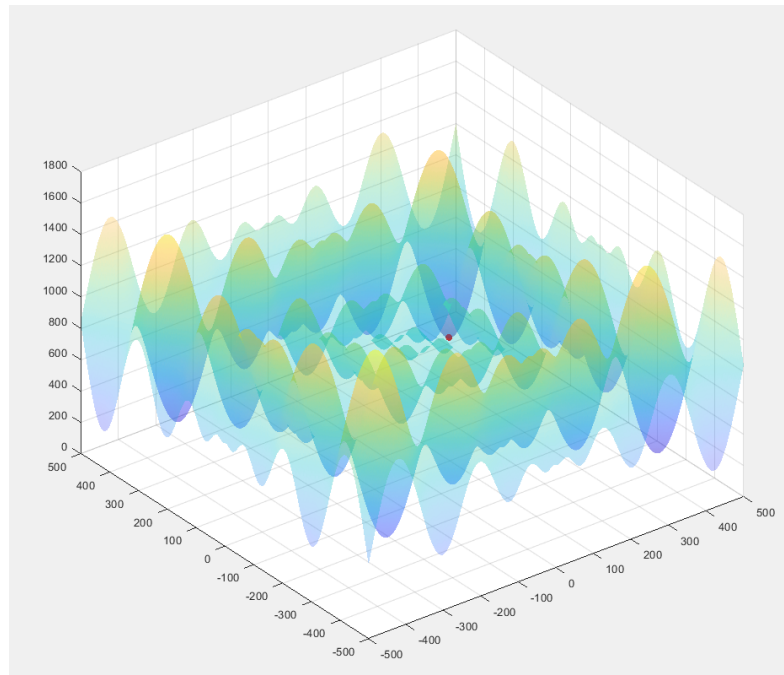
da Função Schwefel é dada pela Equação 12:

$$z = 837,96542 - (x\text{sen}(\sqrt{|x|}) + y\text{sen}(\sqrt{|y|})) \quad (11)$$

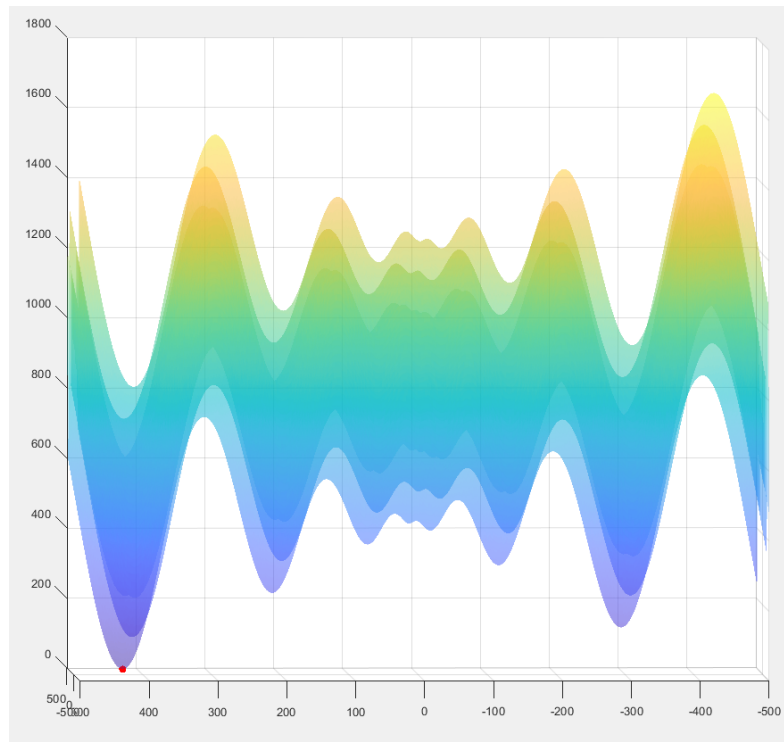
É conhecido que a função possui um ponto de mínimo em $x = 420,9687$, $y = 420,9687$ onde $z = 0$. A partir desta informação iremos utilizar a função de *fitness*

$$\text{fitness} = \frac{1}{1 + z} \quad (12)$$

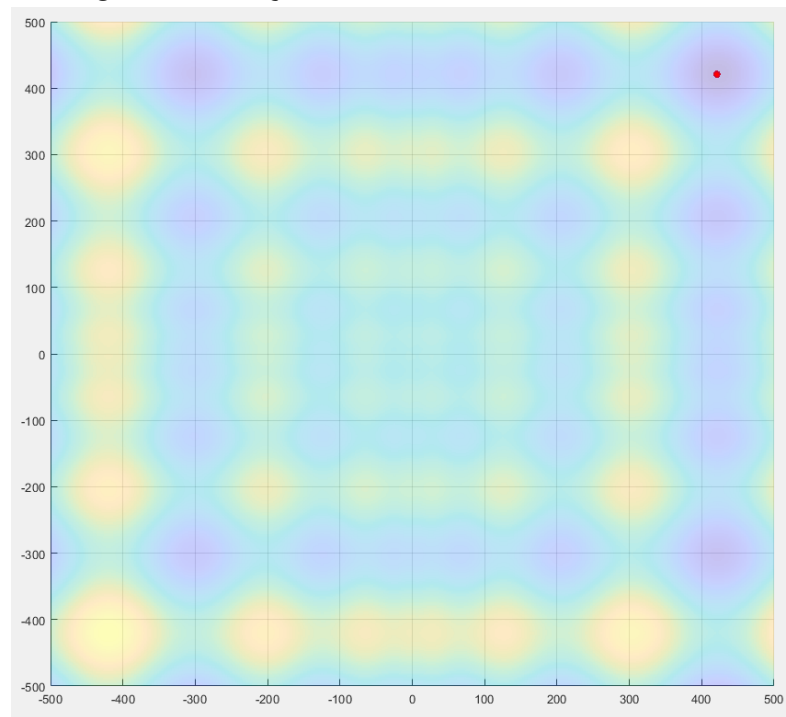
para que possamos encontrar um valor máximo de *fitness* igual a 1 quando minimizarmos a função. Seu ponto de ótimo global pode ser observado nas figuras: Figura 21, Figura 22, Figura 23 e Figura 24.

Figura 21 – Função Schwefel.

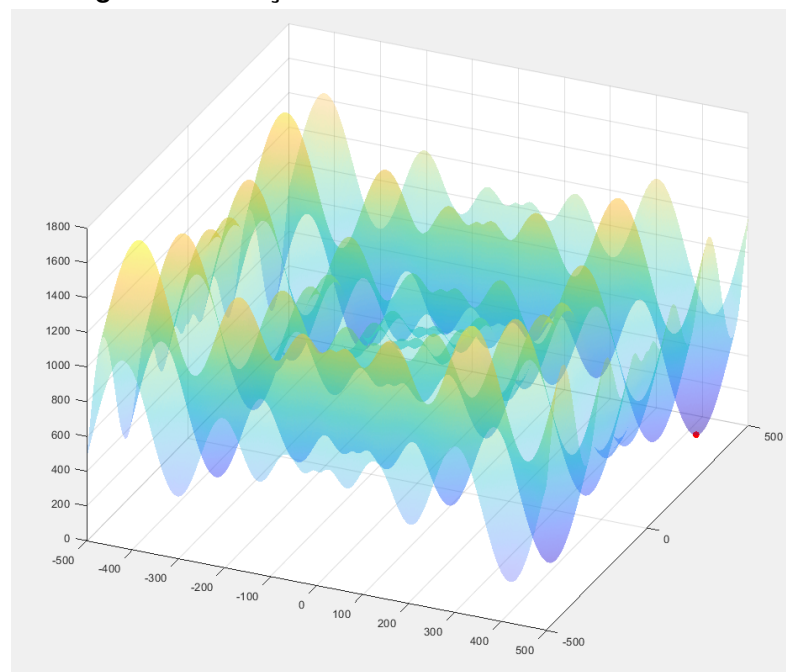
Fonte: Autoria Própria.

Figura 22 – Função Schwefel.

Fonte: Autoria Própria.

Figura 23 – Função Schwefel.

Fonte: Autoria Própria.

Figura 24 – Função Schwefel.

Fonte: Autoria Própria.

6 RESULTADOS E DISCUSSÃO

Este capítulo apresenta os resultados computacionais da otimização das funções *Bechmark* descritas no capítulo 5 por meio dos Algoritmos Genéticos e da Evolução Diferencial.

Para possibilitar a realização da comparação de desempenho entre os diferentes métodos foram realizadas duas baterias de simulações com diferentes propósitos. A primeira verificou a velocidade média que cada estratégia consegue convergir em uma resposta razoável para cada problema. Isto foi feito utilizando um quesito de parada de valor de *fitness* mínimo de 0,95. A segunda foi feita para avaliar a qualidade da resposta de cada função em um número de gerações fixo. O quesito de parada neste caso foi o de 1000 gerações.

Em cada um dos cenários foram realizados 100 testes de otimização independentes para cada combinação de indivíduos para que os testes possuam uma maior consistência, tendo em vista que a aleatoriedade intrínseca dos Algoritmos Evolutivos não afete significativamente as análises das respostas.

Cada simulação da primeira bateria teve um limite de gerações máximo de 15.000. Caso uma simulação não fosse capaz de convergir na resposta desejada neste tempo, é considerado que ela não irá convergir.

As simulações foram feitas utilizando o software *MatLab* e os resultados destas simulações serão apresentados a seguir, separados por função. Observa-se que os resultados apresentados são médias das 100 simulações independentes.

6.1 FUNÇÃO EXPONENCIAL

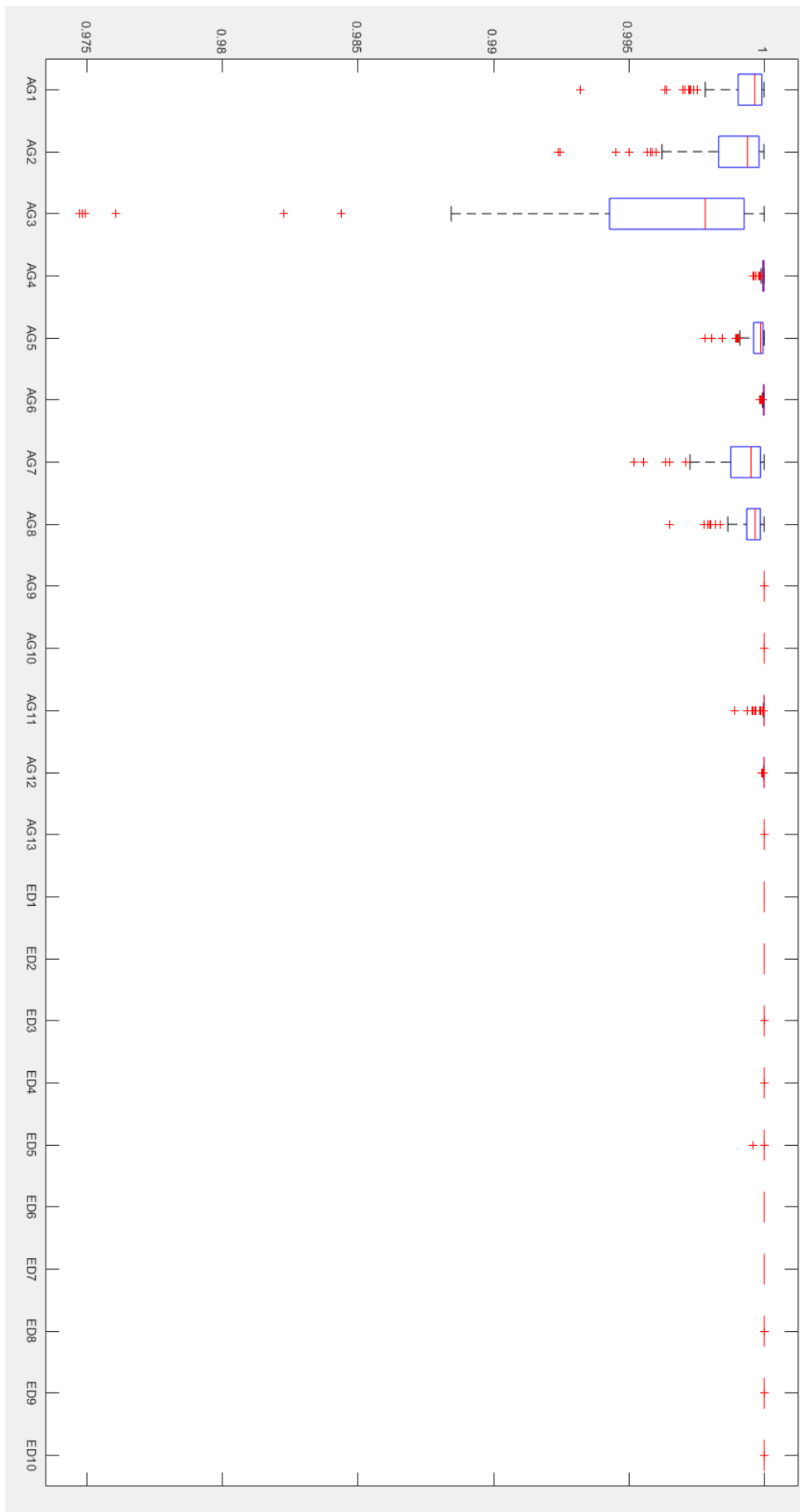
A função exponencial foi otimizada utilizando uma população de 20 agentes ou indivíduos para cada combinação do AG e da ED. O resultado da primeira bateria de simulações para a função exponencial pode ser analisado na Tabela 1 com a melhor estratégia de AG e ED destacados em negrito. O gráfico *boxplot* dos valores de *fitness* da segunda bateria de simulações para a função exponencial pode ser observado na Figura 25. Um *boxplot* é um gráfico estatístico utilizado para avaliar uma distribuição empírica de dados, mostrando o primeiro e terceiro quartil dos dados e sua mediana.

Tabela 1 – Resultados das Simulações da Função Exponencial

Algoritmo	Média de Gerações para Convergência	Taxa de Convergência do algoritmo
AG1	17,07	1,00
AG2	14,72	1,00
AG3	64,84	1,00
AG4	13,82	1,00
AG5	17,50	1,00
AG6	11,67	1,00
AG7	13,25	1,00
AG8	19,49	1,00
AG9	1,00	1,00
AG10	7,29	1,00
AG11	4,58	1,00
AG12	9,62	1,00
AG13	18,66	1,00
ED1	21,93	1,00
ED2	25,39	1,00
ED3	70,79	1,00
ED4	41,61	1,00
ED5	12,88	1,00
ED6	21,94	1,00
ED7	36,39	1,00
ED8	77,17	1,00
ED9	52,97	1,00
ED10	15,13	1,00

Fonte: Autoria própria.

Figura 25 – Boxplot do Fitness da Função Exponencial.



Fonte: Autoria Própria.

6.2 FUNÇÃO RASTRIGIN

Para a função Rastrigin, o tamanho da população foi setado em 30 indivíduos, compatível com a dimensão do espaço de busca. O resultado da primeira bateria de simulações pode ser visto na Tabela 2 com a melhor estratégia de AG e ED destacados em negrito:

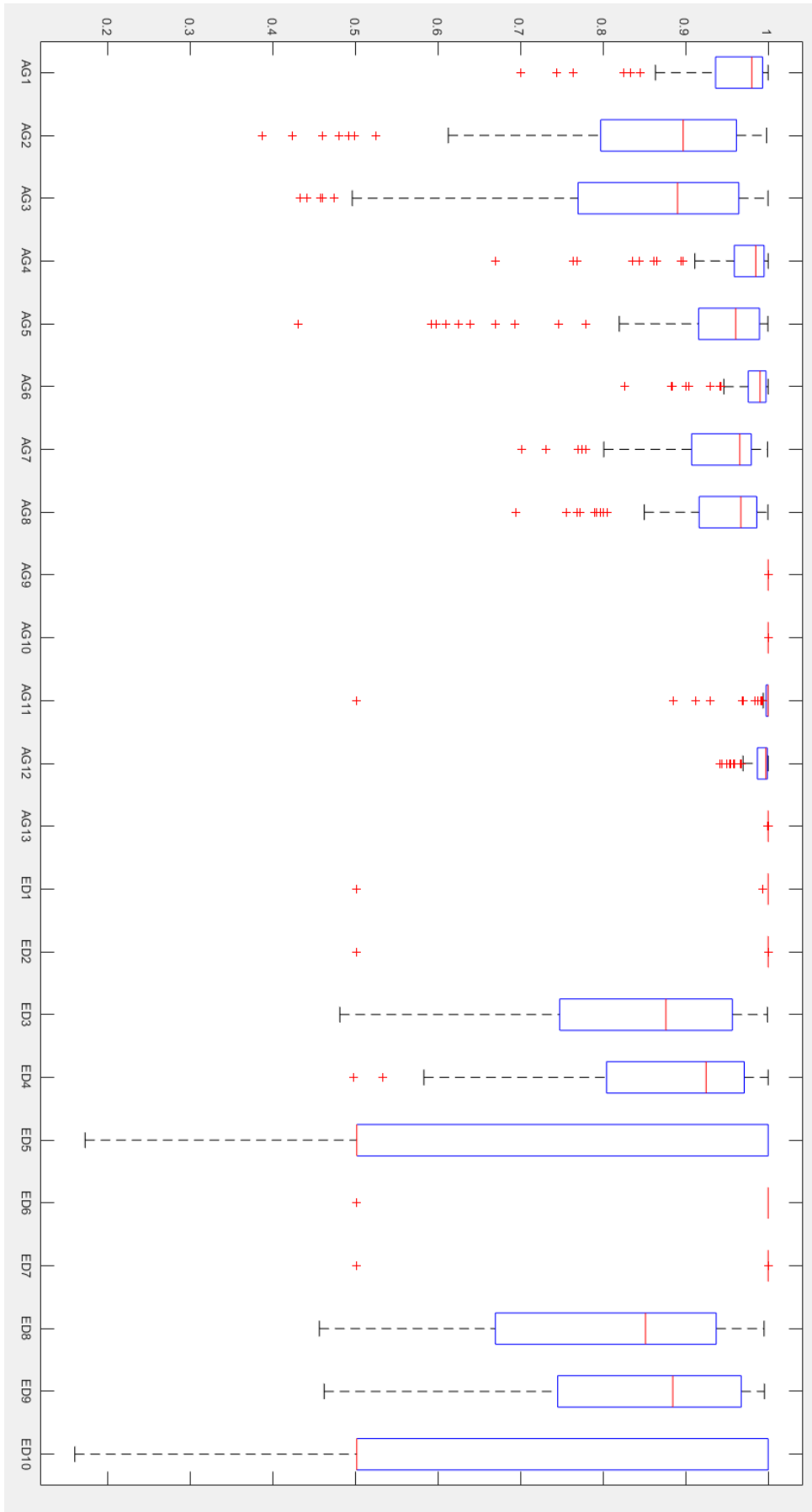
Tabela 2 – Resultados das Simulações da Função Rastrigin

Algoritmo	Média de Gerações para Convergência	Taxa de Convergência do algoritmo
AG1	949,39	1,00
AG2	1142,06	1,00
AG3	3002,62	1,00
AG4	626,02	1,00
AG5	996,73	1,00
AG6	562,21	1,00
AG7	1108,63	1,00
AG8	972,32	1,00
AG9	99,40	1,00
AG10	127,70	1,00
AG11	302,49	1,00
AG12	455,11	1,00
AG13	214,88	1,00
ED1	277,44	0,89
ED2	349,84	0,86
ED3	1325,54	1,00
ED4	1118,04	0,96
ED5	115,86	0,45
ED6	262,70	0,84
ED7	423,79	0,87
ED8	1419,39	1,00
ED9	1249,46	0,97
ED10	143,92	0,38

Fonte: Autoria própria.

O boxplot dos valores de fitness da segunda bateria de simulações é mostrado
Figura 25:

Figura 26 – Boxplot do Fitness da Função Exponencial.



Fonte: Autoria Própria.

6.3 FUNÇÃO SCHWEFEL

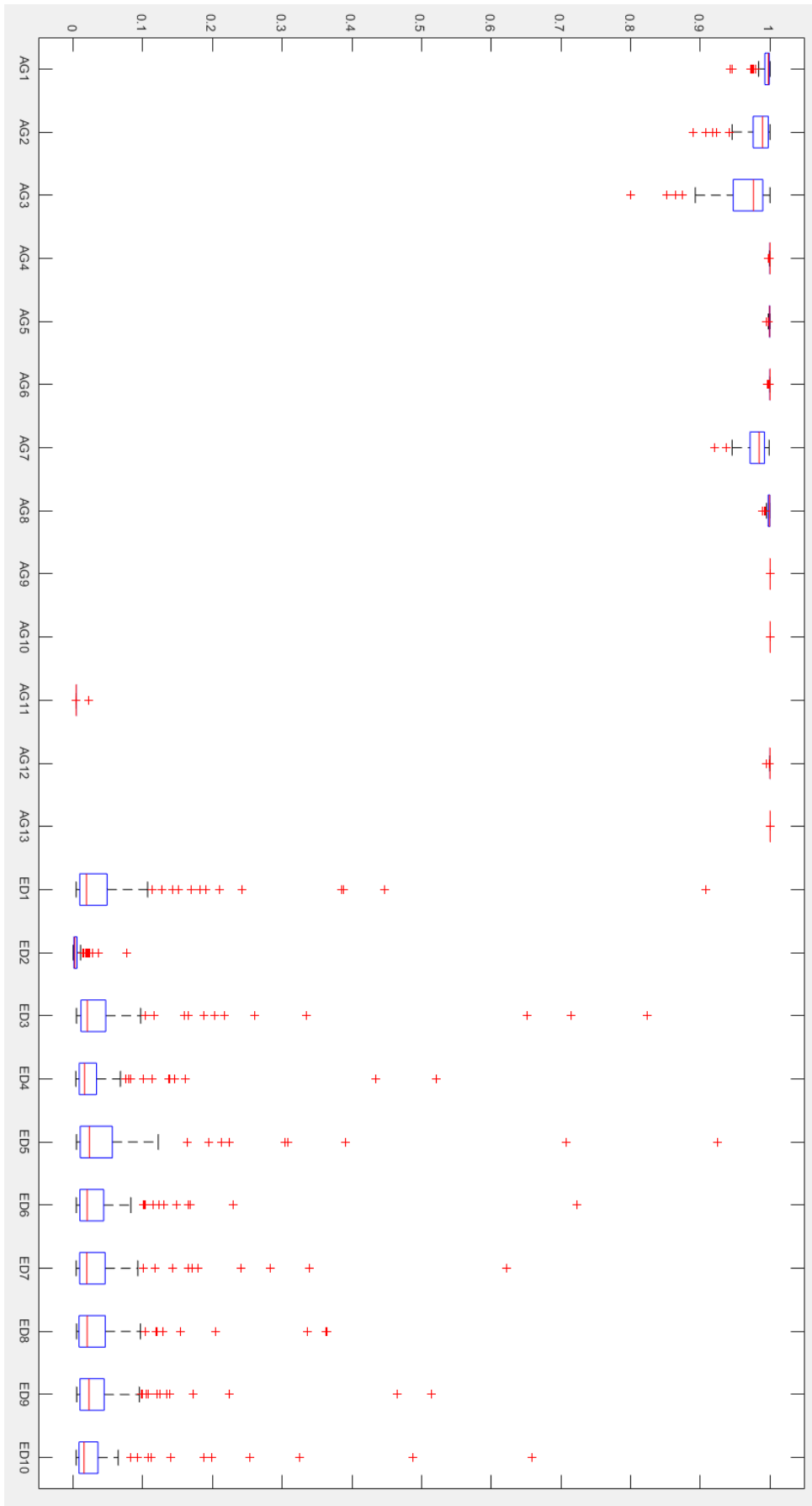
A função Schwefel apresenta espaço de busca muito mais elevado que as demais. dessa forma, foi necessário utilizar uma população de 500 agentes para análise de convergência. O resultado da primeira bateria de simulações pode ser analisado na Tabela 1, enquanto o gráfico boxplot dos valores de fitness da estão presentes na Figura 27 com a melhor estratégia de AG e ED destacados em negrito:

Tabela 3 – Resultados das Simulações da Função Schwefel

Algoritmo	Média de Gerações para Convergência	Taxa de Convergência do algoritmo
AG1	366,36	1,00
AG2	174,05	1,00
AG3	768,17	1,00
AG4	56,96	1,00
AG5	103,62	1,00
AG6	53,46	1,00
AG7	101,61	1,00
AG8	60,33	1,00
AG9	1,02	1,00
AG10	9,66	1,00
AG11	—	0,00
AG12	51,28	1,00
AG13	25,41	1,00
ED1	—	0,00
ED2	—	0,00
ED3	—	0,00
ED4	—	0,00
ED5	—	0,00
ED6	—	0,00
ED7	—	0,00
ED8	—	0,00
ED9	—	0,00
ED10	—	0,00

Fonte: Autoria própria.

Figura 27 – Boxplot do Fitness da Função Schwefel.



Fonte: Autoria Própria.

6.4 DISCUSSÃO DOS RESULTADOS

Pelo volume de dados gerados e simulações realizadas, foi possível fazer múltiplas observações pertinentes considerando os resultados obtidos.

Fazendo uma comparação entre métodos de Seleção observa-se uma velocidade de convergência média mais rápida e um valor de *fitness* médio maior com menor dispersão quando utilizado um método de Torneio quando comparado a Roleta note o comportamento de AG2 e AG4 nas três funções.

Comparando os dois tipos de Roleta utilizados, não foi encontrada uma diferença significativa na velocidade de convergência. Entretanto, nota-se um valor de *fitness* mais alto e com menor dispersão no método de *Stochastic universal sampling*, em detrimento a roleta clássica, como mostram AG2 ao AG7.

Também foi observada uma menor taxa de convergência e valores de *fitness* quando utilizado os pais e filhos na seleção quando comparadas as estratégias AG2 e AG4 com AG3 e AG5. Para o Torneio, nota-se que respostas melhores e mais rápidas quando utilizado o torneio da morte em vez do Torneio Simples. Isso é corroborado pelos resultados obtidos do AG 4 ao AG 6..

O uso da estratégia de *Niching*, levou a uma pior performance dos algoritmos, como se pode ver nos valores no AG4 ao AG8.

Os diferentes métodos de *Crossover*, podemos observar comparando os resultados do AG1 com o AG2, que em ambas funções Exponencial e Schwefel houve um aumento significativo no tempo de convergência, quando utilizada uma taxa de *crossover* não unitária. Porém, em todas as três funções essa estratégia gerou valores de *fitness*, em média, melhores e com menor dispersão.

Sobre as três variações de *crossover*, foi observado que nas duas funções multimodais (Rastrigin e Schwefel) o modelo SBX diminuiu o tempo de convergência médio em mais da metade quando comparado com o *Crossover* de Ponto (AG4 e AG13), alcançando também valores maiores e com menor dispersão de *fitness* máximos em todas as funções.

Já o *Crossover* Aritmético obteve maiores velocidades de convergência e valores de *fitness* quando comparados com o *crossover* de ponto nas duas primeiras funções. Entretanto esta estratégia não foi capaz de alcançar uma resposta ao problema para a função Schwefel em nenhuma das 100 simulações das duas baterias de

testes.

Quanto às duas variações de Mutação, foi observado que a Mutação Dinâmica (AG 12) obteve melhores resultados tanto em velocidade de convergência e valores de *fitness* máximos encontrados quando em detrimento ao mesmo algoritmo com uma taxa de mutação fixa (AG 4).

Foi observado, também, que os algoritmos AG9 e AG 10, os quais utilizam uma estratégia de busca local, foram os com maior velocidade de convergência e melhor valor de *fitness* máximo.

Sobre as avaliações do algoritmo Evolução Diferencial, pode ser observado que, para a Função Exponencial, estratégias que utilizaram apenas uma diferença entre vetores para a mutação (ED1, ED6, ED7 e ED10) obtiveram, em média, respostas mais rápidas do que estratégias que utilizaram duas, como esperado. Foi possível também notar que estratégias com Mutação tipo *Best* (ED5 e ED10) tiveram os melhores resultados em relação as outras estratégias de Mutação e que algoritmos com *Crossover* Binário (ED1 a ED5) apresentaram uma velocidade de convergência maior em comparação com estratégias Exponencial (ED 6 a 10). Não observou-se diferenças grandes nos valores de *fitness*.

Porém, utilizando as variações da ED para as outras Funções *Benchmark* foram encontrados problemas na otimização em um período de tempo razoável.

Para a função Rastrigin, métodos de rápida convergência foram muito suscetíveis a acabar presos em pontos de mínimo local e não encontrar o ponto de ótimo da função, como foi o caso de ED 6 e ED 10. Já as aquelas que foram capazes de obterem taxas de convergência maiores foram significativamente mais lentas para encontrar a resposta, como ED 4 e ED 8.

Para a função Schwefel nenhuma estratégia de Evolução Diferencial foi capaz convergir na resposta desejada dentro dos parâmetros de otimização escolhidos no início. Os valores de *fitness* apresentados anteriormente e no boxplot da função foram encontrados após 15.000 gerações cada, um número muito maior do que o utilizado em qualquer outra estratégia.

Apesar dos problemas encontrados na otimização com Evolução Diferencial, é válido ressaltar que, em todos os momentos que estas estratégias foram capazes de convergir próximas ao ponto de ótimo global. Nesses casos, foram alcançados valores de *fitness* muito próximos ao unitário.

7 CONCLUSÃO

Este trabalho apresentou a inspiração biológica, pseudocódigos e questões relativas à implementação de dois importantes Algoritmos Evolutivos para otimização: Algoritmo Genético (AG) e Evolução Diferencial (ED).

Inspirados no processo natural de evolução Darwiniana e no conceito de seleção natural, tais métodos se destacam na literatura tanto pela grande utilização em problemas reais quanto sua capacidade de resolvê-los de forma adequada.

Nesta investigação foram desenvolvidos computacionalmente 13 variações do AG e outras 10 da ED, variando a forma como seus operadores genéticos são desenvolvidos. Os estudos de caso envolveram a otimização de 3 funções benchmark de 3 dimensões.

A partir dos resultados computacionais coletados, é possível concluir que Algoritmos Evolutivos podem ser excelentes técnicas de otimização, principalmente para problemas não lineares multimodais. De todas as diferentes estratégias apresentadas e testadas, é relevante destacar o fato de que Algoritmos Genéticos foram capazes de gerar respostas para os problemas com consistência, enquanto algoritmos de Evolução Diferencial tiveram problemas de convergência em tarefas multimodais.

Dentro dos Algoritmos Genéticos, as variações que trouxeram melhores resultados foram a Seleção por Torneio da Morte, *Crossover* do Tipo SBX e Mutação Dinâmica. Neste caso, a possibilidade de utilizar uma Taxa de *Crossover* não unitária pode melhorar o tipo de resposta encontrada, mas com um possível aumento de tempo de simulação.

As melhores *performances* gerais encontradas foram utilizando um algoritmo híbrido entre Algoritmos Genéticos e Busca Local, o que já era esperado.

Apesar dos problemas já mencionados com a Evolução Diferencial, acredita-se que mudanças nestes algoritmos possam gerar novas estratégias alternativas altamente eficientes e precisas.

Conclui-se também que mais estudos devem ser feitos na área de otimização Metaheurística e Algoritmos Evolutivos. Trabalhos futuros podem ser desenvolvidos no sentido de comparar o desempenho destas técnicas com modelos de inteligência de enxame, como o PSO e outras funções Benchmark.

REFERÊNCIAS

BACK, Thomas; FOGEL, David B.; MICHALEWICZ, Zbigniew (Ed.). **Handbook of Evolutionary Computation**. 1st. ed. Bristol, UK, UK: IOP Publishing Ltd., 1997. ISBN 0750303921. Citado 3 vezes nas páginas 21, 23 e 25.

DARWIN, C. **Origem das espécies**. [S.l.]: Lello & Irmão Editores, 1959. (Biblioteca racionalista). ISBN 9788537004531. Citado na página 16.

DAWKINS, R.; FLORSHEIM, G.H.M. **O gene egoísta**. [S.l.]: Ed Itatiaia, 2001. (O homem e a ciência). ISBN 9788531901881. Citado na página 20.

DEB, Kalyanmoy; KUMAR, Amarendra. Real-coded genetic algorithms with simulated binary crossover: Studies on multimodal and multiobjective problems. **Complex Systems**, v. 9, 1995. Citado na página 28.

ELMIHOUB, T. et al. Performance of hybrid genetic algorithms incorporating local search. In: **in 18th European Simulation Multiconference (ESM2004)**. [S.l.: s.n.], 2004. p. 154–160. Citado na página 30.

GOLDBERG, David E. **Genetic Algorithms in Search, Optimization and Machine Learning**. 1st. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0201157675. Citado 4 vezes nas páginas 21, 22, 23 e 25.

_____. Genetic and evolutionary algorithms come of age. **Commun. ACM**, ACM, New York, NY, USA, v. 37, n. 3, p. 113–119, mar. 1994. ISSN 0001-0782. Citado na página 28.

GRIFFITHS, A.J.F. et al. **Introduction to Genetic Analysis + Ebook**. [S.l.]: Macmillan Higher Education, 2008. ISBN 9781429235037. Citado 3 vezes nas páginas 18, 19 e 20.

HEULE, Marijn; KULLMANN, Oliver; MAREK, Victor W. **Solving and Verifying the Boolean Pythagorean Triples Problem via Cube-and-Conquer**. 2016. 228-245 p. Citado na página 13.

HOLLAND, J.H. **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence**. [S.l.]: University of Michigan Press, 1975. ISBN 9780472084609. Citado 2 vezes nas páginas 21 e 27.

HONG, Tzung-Pei; WANG, Hong-Shung. A dynamic mutation genetic algorithm. In: **1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No.96CH35929)**. [S.l.: s.n.], 1996. v. 3, p. 2000–2005 vol.3. ISSN 1062-922X. Citado na página 29.

LAMARCK, J.B. de Monet de. **Filosofia zoológica**. [S.l.]: La Nuova Italia, 1976. (Pensatori antichi e moderni, v. 1). Citado na página 16.

LIN, Chuan; QING, Anyong; FENG, Quanyuan. A comparative study of crossover in differential evolution. **Journal of Heuristics**, v. 17, n. 6, p. 675–703, Dec 2011. ISSN 1572-9397. Citado na página 35.

MAHFOUD, Samir W. **Niching Methods for Genetic Algorithms**. 1995. Tese (Doutorado), Champaign, IL, USA, 1995. UMI Order No. GAX95-43663. Citado na página 30.

MENDEL, G.; MANGELSDORF, P.C. **Experiments in Plant Hybridisation**. [S.l.]: Harvard University Press, 1965. ISBN 9780674278004. Citado 2 vezes nas páginas 17 e 19.

MICHALEWICZ, Zbigniew. Evolution strategies and other methods. In: _____. **Genetic Algorithms + Data Structures = Evolution Programs**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996. p. 159–177. ISBN 978-3-662-03315-9. Citado 4 vezes nas páginas 21, 22, 23 e 24.

MITCHELL, Melanie. **An Introduction to Genetic Algorithms**. Cambridge, MA, USA: MIT Press, 1996. ISBN 0-262-13316-4. Citado na página 22.

MOORE, G. E. Cramming more components onto integrated circuits, reprinted from electronics, volume 38, number 8, april 19, 1965, pp.114 ff. **IEEE Solid-State Circuits Society Newsletter**, v. 11, n. 3, p. 33–35, Sept 2006. ISSN 1098-4232. Citado na página 13.

MUKHERJEE, S. **The Gene: An Intimate History**. [S.l.]: Bodley Head, 2016. ISBN 9781847922632. Citado 2 vezes nas páginas 17 e 20.

STORN, Rainer; PRICE, Kenneth. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. **Journal of Global Optimization**, v. 11, n. 4, p. 341–359, Dec 1997. ISSN 1573-2916. Citado 2 vezes nas páginas 21 e 32.

TUSAR, Tea; FILIPIC, Bogdan. **Differential Evolution versus Genetic Algorithms in Multiobjective Optimization**. 2007. 257-271 p. Citado na página 32.

VOß, Stefan. Meta-heuristics: The state of the art. In: **Proceedings of the Workshop on Local Search for Planning and Scheduling-Revised Papers**. Berlin, Heidelberg: Springer-Verlag, 2001. (ECAI '00), p. 1–23. ISBN 3-540-42898-4. Citado na página 13.

ZHONG, Jinghui et al. Comparison of performance between different selection strategies on simple genetic algorithms. **International Conference on Computational Intelligence for Modelling Control and Automation**, 2005. Citado 2 vezes nas páginas 26 e 27.