

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

WALACE RUTIELO LOPES SANTOS

ALGORITMOS EVOLUTIVOS PARA OTIMIZAÇÃO BINÁRIA

TRABALHO DE CONCLUSÃO DE CURSO

**PONTA GROSSA
2019**

WALACE RUTIELO LOPES SANTOS

ALGORITMOS EVOLUTIVOS PARA OTIMIZAÇÃO BINÁRIA

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

Orientador: Prof. Dr. Hugo Valadares Siqueira

PONTA GROSSA

2019

Ficha catalográfica elaborada pelo Departamento de Biblioteca da
Universidade Tecnológica Federal do Paraná, Câmpus Ponta Grossa
n. 00/2019

A000 Santos, Wallace Rutielo Lopes

Algoritmos Evolutivos para otimização binária / Wallace Rutielo Lopes Santos. – Ponta Grossa, PR: [s.n.], 2019.
59 f. : il. ; 30 cm.

Orientador: Prof. Dr. Hugo Valadares Siqueira

Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica)
– Departamento de Elétrica. Universidade Tecnológica Federal do Paraná.
Ponta Grossa, 2019.

1. Otimização. 2. Algoritmos Evolutivos. 3. Algoritmo Genético. 4. Evolução Diferencial. 5. Função Benchmark. I. Siqueira, Hugo Valadares. II. Universidade Tecnológica Federal do Paraná, Câmpus Ponta Grossa. III. Título.

CDD 000.00



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Ponta Grossa
Diretoria de Graduação e Educação Profissional
Departamento Acadêmico de Eletrônica
Bacharelado em Engenharia Elétrica



TERMO DE APROVAÇÃO

ALGORITMOS EVOLUTIVOS PARA OTIMIZAÇÃO BINÁRIA

por

WALACE RUTIELO LOPES SANTOS

Este Trabalho de Conclusão de Curso foi apresentado em 06 de dezembro de 2019 como requisito parcial para a obtenção do título de Bacharel(a) 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(a). Dr. Hugo Valadares Siqueira
Orientador(a)

Prof(a). Dr^a. Fernanda Cristina Côrrea
Membro Titular

Prof(a). Dr. Eduardo Tadeu Bacalhau
Membro Titular

Prof. Dr. Josmar Ivanqui
Responsável pelos TCC

Prof. Dr. Sergio Okida
Coordenador do Curso

– O Termo de Aprovação assinado encontra-se na Coordenação do Curso –

Dedico este trabalho a minha família e aos
meus amigos, pelos momentos de
ausência.

AGRADECIMENTOS

Primeiramente agradeço à Deus por toda a força durante essa caminhada.

Aos meus pais, Lúcia e João, e à minha irmã, Glauce, que depositaram sua confiança e apoio permitindo que eu chegasse até aqui.

Aos irmãos do Nerdarada, Biuk, Félipe, Duds, Manoer, Guimerme, Rutz e Migolé por todo o suporte e pelas risadas.

Ao meu orientador Hugo, um grande amigo, por toda a paciência e ajuda durante o desenvolvimento deste trabalho.

Aos amigos do LICON pelo companheirismo.

Enfim, a todos os que de alguma forma contribuíram para a realização deste trabalho.

Um pássaro que repousa numa árvore
nunca teme que o galho quebre, porque a
sua confiança não é no galho, mas nas
suas próprias asas. (BUSANELLO, Elis).

RESUMO

SANTOS, Wallace Rutiello Lopes. **Algoritmos Evolutivos para otimização binária**. 2019. 59 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2019.

Este trabalho tem por objetivo aplicar métodos bio-inspirados de otimização à problemas de natureza binária, muito presentes na literatura. Busca-se analisar o desempenho dos métodos a partir de dois algoritmos evolutivos: o algoritmo genético e a evolução diferencial. Estudos de casos são propostos com a finalidade de obter uma análise comparativa de desempenho dos algoritmos. Os estudos propõe a execução de 10 variações para cada algoritmo, aplicados a duas funções Benchmark: *OneMax Problem* e o Problema da Mochila. Os resultados computacionais são discutidos, destacando-se as boas soluções obtidas pelo algoritmo de evolução diferencial binária.

Palavras-chave: Otimização. Algoritmos Evolutivos. Algoritmo Genético. Evolução Diferencial. Função Benchmark.

ABSTRACT

SANTOS, Wallace Rutiello Lopes. **Evolutionary Algorithms for binary optimization**. 2019. 59 p. Final Coursework (Bachelor's Degree in Eletrical Engineering) – Federal University of Technology – Paraná. Ponta Grossa, 2019.

This work aims to apply bio-inspired methods to the optimization of problems of binary nature, very present in the literature. Research and analyze the performance of methods from two evolutionary algorithms: the genetic algorithm and a differential evolution. Case studies are proposed by obtaining a comparative analysis of the performance of the algorithms. The studies propose the execution of 10 variations for each algorithm, using the two benchmark functions: OneMax Problem and Knapsack Problem. The computational results are discussed, highlighting as good solutions caused by the binary differential evolution algorithm.

Keywords: Optimization. Evolutionary Algorithms. Genetic Algorithm. Differential Evolution. Benchmark Function.

LISTA DE ALGORITMOS

Algoritmo 1 – Algoritmo Genético.	26
Algoritmo 2 – Evolução Diferencial.	31

LISTA DE ILUSTRAÇÕES

Figura 1 – Variedades de bicos nas aves	17
Figura 2 – Cruzamento de ervilhas	18
Figura 3 – Estrutura do DNA	19
Figura 4 – Recombinação de um cromossomo	19
Figura 5 – Roleta clássica (a), <i>Universal Stochastic Sampling</i> (b)	22
Figura 6 – Exemplo de seleção por torneio	23
Figura 7 – <i>Recombinação binária para o AG.</i>	24
Figura 8 – Mutação binária.	25
Figura 9 – Fluxograma Algoritmo Genético.	26
Figura 10 – Recombinação da Evolução Diferencial.	30
Figura 11 – Fluxograma Evolução Diferencial.	31
Figura 12 – Boxplot do Onemax Problem para 50 dimensões.	40
Figura 13 – Boxplot do Onemax Problem para 100 dimensões.	41
Figura 14 – Boxplot do Onemax Problem para 250 dimensões.	42
Figura 15 – Boxplot do Onemax Problem para 500 dimensões.	43
Figura 16 – Boxplot para mochila de 30 dimensões.	46
Figura 17 – Boxplot para mochila de 100 dimensões.	48
Figura 18 – Boxplot para mochila de 250 dimensões.	50
Figura 19 – Boxplot para mochila de 500 dimensões.	52

LISTA DE TABELAS

Tabela 1 – Exemplo de avaliação para a roleta	23
Tabela 2 – Cálculo de <i>fitness</i> para <i>OneMax Problem</i>	34
Tabela 3 – Tabela de itens com seus respectivos valores e pesos	35
Tabela 4 – Resolução para o problema da mochila	35
Tabela 5 – Características para cada variação da mochila	36
Tabela 6 – Média de convergência do <i>fitness</i> para cada dimensão do <i>OneMax Problem</i>	38
Tabela 7 – Média de convergência para 30 dimensões do problema da mochila	44
Tabela 8 – Média de convergência para 100 dimensões da mochila	47
Tabela 9 – Média de convergência para 250 dimensões <i>Knapsack Problem</i> . .	49
Tabela 10 – Média de convergência para 500 dimensões da mochila	51

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	14
1.2	ORGANIZAÇÃO DO TRABALHO	14
2	BASE BIOLÓGICA	16
2.1	TEORIA DA EVOLUÇÃO	16
2.2	GENÉTICA E EVOLUÇÃO	17
2.3	ALGORITMOS EVOLUTIVOS	20
3	ALGORITMO GENÉTICO	21
3.1	SELEÇÃO	22
3.2	RECOMBINAÇÃO	24
3.3	MUTAÇÃO	25
3.4	<i>VARIANTES DO ALGORITMO GENÉTICO</i>	25
4	EVOLUÇÃO DIFERENCIAL	28
5	FUNÇÃO BENCHMARK	33
5.1	<i>ONEMAX PROBLEM</i>	33
5.2	<i>PROBLEMA DA MOCHILA</i>	34
6	RESULTADOS E DISCUSSÃO	37
6.1	<i>FUNÇÃO ONEMAX PROBLEM</i>	37
6.2	<i>PROBLEMA DA MOCHILA</i>	44
6.2.1	30 dimensões para problema da mochila	44
6.2.2	100 dimensões para problema da mochila	47
6.2.3	250 dimensões para problema da mochila	49
6.2.4	500 dimensões para problema da mochila	51
6.3	DISCUSSÃO DOS RESULTADOS	53
7	CONCLUSÃO	55
	REFERÊNCIAS	56

1 INTRODUÇÃO

Os humanos são seres que possuem consciência da sua própria evolução em relação aos outros seres vivos da Terra. Com o tempo, percebeu-se necessidades de alimentação, sobrevivência e como resolvê-las de modo prático e rápido. Com isso, desenvolveu-se técnicas e ferramentas para poder atender suas necessidades e resolver os problemas encontrados (CAMPOS; SANTOS; XAVIER, 1997). Descobertas e criações que foram pontos de inflexão para a humanidade melhoraram o modo como vivemos definindo o rumo da história, desde a possibilidade de manipulação do fogo, criação de ferramentas básicas tais como: a roda, energia elétrica, transistores, computadores, internet e etc (FARIAS; SELITTO, 2011).

As dificuldades encontradas eram cada vez mais complexas e necessitavam de melhores recursos e técnicas para poder encontrar uma solução. As respostas para problemas do dia a dia podem ser encontradas na natureza, observando o comportamento, padrões, a maneira como funciona e entender, o que levou a chegar naquele modo que está hoje. A natureza também tem seus problemas e as soluções vieram através da evolução. Seguindo a mesma lógica e adaptando para a nossa realidade, poderíamos resolver os nossos ou chegar a bons resultados (CASTRO, 2006).

Problemas que com a tecnologia atual poderiam levar anos para alcançar um resultado, manifestou-se a necessidade de técnicas que poderiam chegar a resultado satisfatório. Uma classe dessas técnicas são as ferramentas de otimização, as quais abordam programação dinâmica, programação linear e as metaheurísticas que englobam redes neurais, sistemas Fuzzy e algoritmos bio-inspirados como algoritmo genético, otimização por enxame de partículas (PSO), evolução diferencial e etc (KANNAN; SLOCHANAL; PADHY, 2005).

A Computação Natural (inspirada na natureza) aparece como uma alternativa às soluções já existentes para determinados problemas, isto é, a sua aplicação é mais conveniente quando se possui um problema de alta complexidade, com grande quantidade de variáveis e possíveis soluções. Uma vez que o problema não necessite desse tratamento, provavelmente ele possui alguma solução específica (LINDEN, 2008).

Na computação evolucionária são desenvolvidas técnicas de busca e otimização inspiradas na evolução biológica para solucionar tarefas das mais diversas

áreas do conhecimento, como engenharia, computação, física, dentre outros (LINDEN, 2008). Os algoritmos evolutivos são ferramentas que emulam os processos naturais de evolução das espécies como seleção, reprodução e mutação, para resolverem os citados problemas.

Uma instância relevante da ciência é a otimização no domínio binário, a qual envolve problemas do tipo on/off, ou seleção de variáveis. Isto se torna particularmente importante com a disponibilidade de dados e armazenagem que hoje o mundo dispõe. O Algoritmo Genético binário e a Evolução Diferencial em suas versões binárias e booleanas são os algoritmos evolutivos abordados neste trabalho (HOLLAND et al., 1992).

1.1 OBJETIVOS

O objetivo geral deste trabalho é conduzir uma análise sobre otimização com algoritmos evolutivos, a saber algoritmos genéticos e evolução diferencial, utilizando funções *Benchmark*. A proposta é gerar parâmetros de comparação e avaliar a qualidade dos resultados, propiciando uma análise geral de desempenho das propostas.

Este objetivo pode alcançado por meio dos seguintes objetivos específicos:

- Detalhar os tipos de algoritmos abordados;
- Implementar as variações de algoritmos evolutivos e validá-las;
- Avaliar as propostas em funções *Benchmark*, como o OneMax Problem e o Problema da Mochila, com variado número de dimensões;
- Apresentar os resultados computacionais obtidos, verificando os benefícios e desvantagens para cada função definida.

1.2 ORGANIZAÇÃO DO TRABALHO

No segundo capítulo deste trabalho, tem-se uma revisão dos conceitos da evolução das espécies, que é a base biológica que fundamenta os Algoritmos Evolutivos. As variações e como funciona o Algoritmo Genético serão abordados no capítulo 3.

No quarto capítulo serão analisadas as propriedades da Evolução Diferencial e as diferentes estratégias implementadas. Para o Capítulo 5 serão apresentadas duas Funções *Benchmark* utilizadas para testes e avaliação de performance.

No sexto capítulo são apresentados os resultados obtidos das simulações realizadas para os testes e feita uma discussão estes. No último Capítulo é apresentada a conclusão do trabalho.

2 BASE BIOLÓGICA

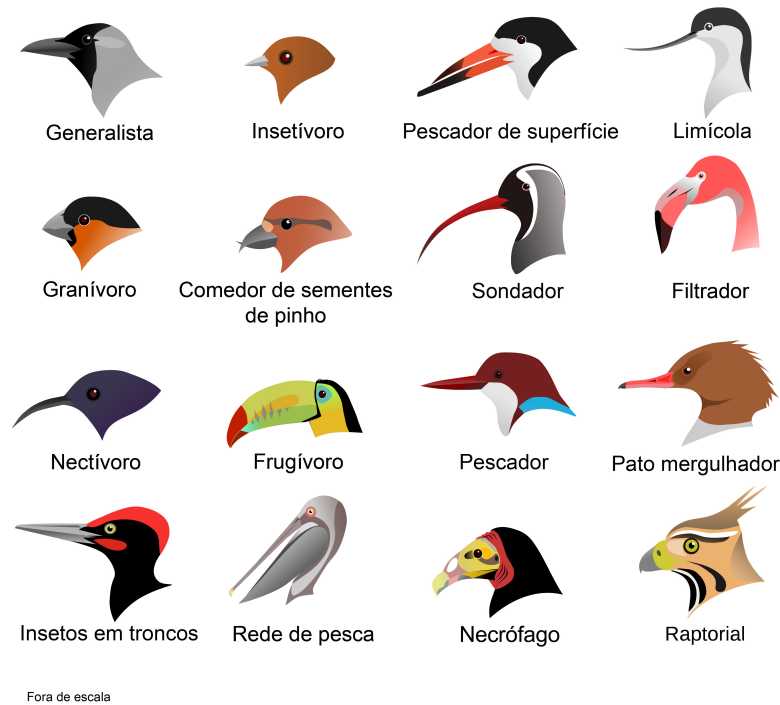
2.1 TEORIA DA EVOLUÇÃO

A primeira teoria ampla no século XIX exposta por Jean Baptiste Lamarck, um naturalista francês, sustentava que os organismos vivos apresentavam um desenvolvimento, sendo os seres humanos a forma de maior progressão. Os organismos evoluem através do tempo, das formas mais simples e inferiores para as mais complexas. Conforme os seres se adaptam aos ambientes em que vivem, ocorrem alterações. Por exemplo, quanto mais um órgão é usado durante a vida, mais reforçado ficará, da mesma forma que o menos utilizado com o tempo deixaria de existir. Esta característica que se desenvolveu ou desapareceu, é herdada pelas próximas gerações, hipótese conhecida como herança de características adquiridas ou lamarckismo. As contribuições de Lamarck foram importantes para entender a evolução biológica das espécies (VON; FERNANDO, 2000).

Em 1859, a teoria da evolução apresentada no livro "A Origem das Espécies" de Charles Darwin, propõe que todos os seres vivos possuem um ancestral comum e as espécies evoluem. O autor sugere a seleção natural como um recurso para a evolução, em que características são repassadas para a próxima geração de forma que colaboram para a reprodução e sobrevivência da espécie (DARWIN; MIRANDA; BIZZO, 2019).

Darwin, em suas viagens para observar e estudar as espécies nas Ilhas de Galápagos, uma de suas viagens mais conhecidas, observou diversas espécies, sendo a de tentilhões a obter um maior destaque. Isto lhe despertou particular interesse em entender como as espécies se originaram. Devido a cada espécie ter se isolado em ilhas diferentes do arquipélago, em cada uma se pode perceber que este grupo de pássaros, dependendo das condições do meio onde viviam, exibiam características diferentes. Logo, um teria relação com o outro, sendo o meio um fator que influenciava sobre a particularidade de sua biologia. Um desses fatores era o alimento disponível em cada ilha. Assim, cada pássaro tinha um bico com tamanhos e formatos diferentes, destinado para alcançar e comer os alimentos que encontrava. Na Figura 1, pode-se verificar os diversos tipos de bicos de pássaros encontrados na natureza.

Figura 1 – Variedades de bicos nas aves



Fonte: www.quora.com/Do-birds-eat-rice

Outro naturalista da mesma época, Alfred Russel Wallace, de forma independente, alcançou resultados muito semelhantes aos de Darwin. Em 1858, Wallace enviou suas anotações ao colega de área que na época desenvolvia a sua teoria de maneira detalhada desde 1850. Darwin notou através desse manuscrito a similaridade com as suas pesquisas. Ambos compreendiam que o aumento da população realça a disputa por recursos e aquele que sobreviver a esta competição aumentariam suas chances de reprodução possibilitando ter mais herdeiros, sendo esse um fundamento da seleção natural. Isso posteriormente resultou em publicações de trabalhos em conjunto (DARWIN; MIRANDA; BIZZO, 2019).

2.2 GENÉTICA E EVOLUÇÃO

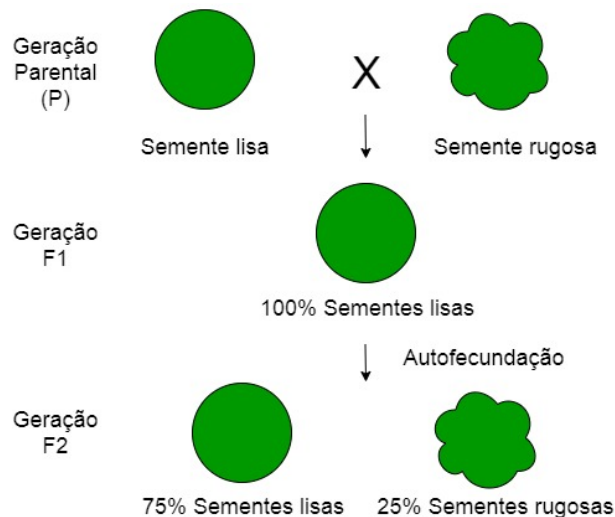
A humanidade busca desde a antiguidade entender de que maneira acontece a sucessão das características de um ser para o seu descendente. Afirmava-se que os filhos apresentavam semelhanças com os pais e que essas eram, de maneira simples, a compreensão sobre herança.

Estudos com viés científico nesta área tiveram início em 1866 através dos experimentos propostos por um monge chamado Gregor Mendel, que publicou um tra-

balho sobre essa leis que abordam a herança e ficou conhecido como pai da genética devido a sua contribuição (HOLLAND et al., 1992).

A princípio Mendel observou atributos distintos em ervilhas, o que o estimulou a entender e estudar o motivo delas apresentarem essas tais diferenças, como pode ser observado na Figura 2. Com o desenvolvimento de suas pesquisas, ele buscava compreender o que ocorria para que o cruzamento entre híbridos de ervilhas concebesse sucessores tão diferentes. Mendel tinha a intenção de criar maneiras de preservar características importantes para a agricultura através de plantas híbridas (CASTRO, 2006).

Figura 2 – Cruzamento de ervilhas

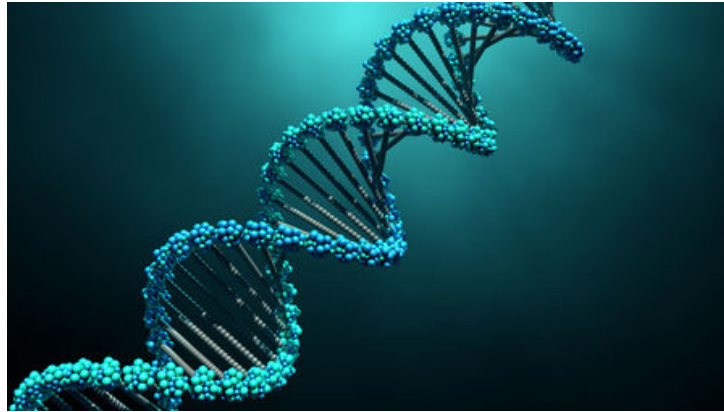


Fonte: Adaptado de LOPES e ROSSO (2005)

Atualmente, a área da Biologia responsável pelo estudo sobre hereditariedade é a genética, que compreende como a sucessão das características são transmitidas a cada geração (CASTRO, 2006). Um dos grandes avanços que este campo proporcionou nas últimas décadas foi a descoberta da estrutura que carrega as informações genéticas, o DNA (Ácido Desoxirribonucleico). Esta é a estrutura chave que transporta as informações fundamentais de cada indivíduo, tendo formato de uma fita helicoidal, coordenando as atividades do corpo, como produção de proteínas, novas células e o processo para absorver os alimentos.

O DNA (Figura 3) nos humanos tem um total de 46 cromossomos, agrupados em 23 pares, sendo composto por cadeias com milhares de genes. Estes definem o seguimento genético no qual estão armazenadas as particularidades de cada indivíduo, desde os atributos externos, como as características físicas, até as funções

Figura 3 – Estrutura do DNA

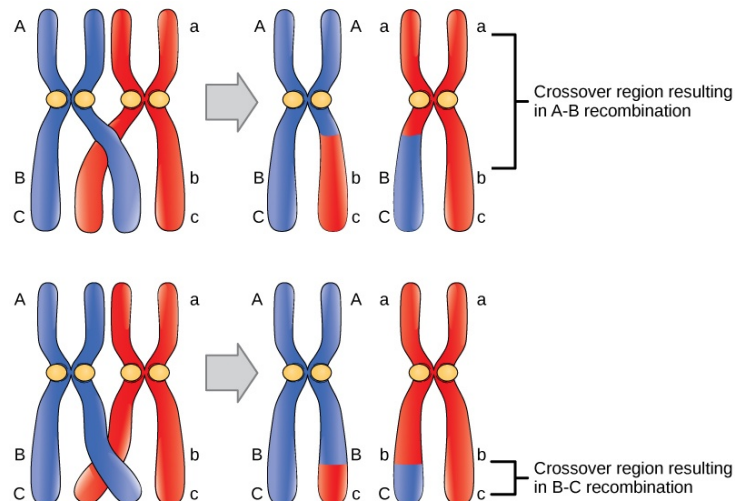


Fonte: hms.harvard.edu/news/function-form

internas das células. Os gametas, células reprodutivas, diferentes das demais por possuírem 23 cromossomos (GRIFFITHS et al., 2006).

As informações genéticas são transmitidas dos pais aos herdeiros por meio da combinação de seus gametas. Entretanto, uma etapa importante que ocorre durante a fecundação é o *crossover* (Figura 4). No decorrer da reprodução ocorre a troca de material genético entre os pares de cromossomos homólogos, sucedendo em dois novos, cada um tendo partes dos cromossomos dos pais (CASTRO, 2006).

Figura 4 – Recombinação de um cromossomo



Fonte: biology.oer.hawaii.edu/biology171/chapter/mapping-genomes/

Essa mescla de codificação genética é importante por gerar uma diversidade de informação genética, atribuindo uma vantagem para a espécie. Desse modo, essa etapa tem sua relevância para que a prole carregue a combinação de características procedentes dos 2 pares de cromossomo dos pais, um da mãe e outro do pai.

Na formação de cada indivíduo, é importante definir os termos genótipo e fenótipo. A definição de genótipo abrange desde cada par de alelos até o conjunto total de genes de um indivíduo. Um determinado genótipo é passado dos pais aos filhos, tendo a capacidade de expressar um fenótipo. Dependendo do nível de interação com o meio, pode-se expressar mais de um fenótipo a partir do mesmo genótipo (LOPES; ROSSO, 2005).

Um fator que é hereditário e que pode ser observado é o caso da miopia. Essa doença acontece quando a luz que adentra os olhos deveria focar na retina, mas acaba se projetando antes. Para a pessoa que sofre deste mal, o que ocorre na prática é que, a imagem mal focada distorce objetos, os quais irão aparentar que estão borrados e levando à necessidade do uso de lentes corretivas para melhorar a visão. O fator genético que provém dos pais não garante que os filhos irão apresentar esses problemas, mas se o pais o manifestam, os filhos estão mais propensos à essa doença (SINGH; TYAGI, 2018).

Outro processo importante para a genética na evolução é a concepção de mutação. No momento da divisão celular, ocorre a duplicação de cromossomos e de maneira aleatória, podem acontecer erros durante a construção dos cromossomos. A maioria das mutações por muitas vezes são irrelevantes, mas algumas impactam de forma positiva contribuindo para a evolução da espécie e, ao longo das gerações, essa alteração estará cada vez mais presente na população (GRIFFITHS et al., 2006).

2.3 ALGORITMOS EVOLUTIVOS

Os algoritmos evolutivos são ferramentas de busca e otimização inspirados na natureza, que emulam os processos naturais de evolução das espécies como a seleção e os operadores genéticos como reprodução e mutação. Os candidatos à solução são os indivíduos, esses sendo apresentados como vetores (LINDEN, 2008). Entre os algoritmos evolutivos tem-se o Algoritmo Genético (HOLLAND et al., 1992) e a Evolução Diferencial (PRICE; STORN; LAMPINEN, 2006).

3 ALGORITMO GENÉTICO

Em 1975, John H. Holland propôs os Algoritmos Genéticos (AG). Inicialmente foram desenvolvidos em codificação binária devido a tecnologia computacional disponível na época, porém o objetivo era a otimização real (HOLLAND et al., 1992).

O AG é inspirado na teoria da evolução de Darwin, na qual os indivíduos mais fortes e adaptados têm maiores chances de transmitir sua carga genética para as próximas gerações. É uma metaheurística de busca e otimização, que tem como propósito encontrar o melhor indivíduo para uma determinada situação (GRIFFITHS et al., 2006), sendo cada indivíduo da população uma solução candidata.

O AG se inicia com indivíduos aleatoriamente gerados, de modo que cada um carrega suas características próprias da mesma forma que um cromossomo. Este é formado por uma cadeia de genes, com várias posições, sendo cada uma denominada de locus, o qual contém um alelo. No caso de uma cadeia binária, esse valor é 0 ou 1. O indivíduo é avaliado de acordo com a função *fitness*, sendo tal avaliação realizada a cada geração. O *fitness* é um valor de aptidão que se atribui ao indivíduo de acordo com o valor da função objetivo, a qual cria a associação entre o problema de otimização e o algoritmo (KONZEN et al., 2003).

A etapa seguinte é a seleção dos indivíduos. O valor de *fitness* que cada um recebeu será um fator determinante para ser selecionado. Após feita a seleção, eles são separados em pares, que representam os pais. No processo da evolução, tem-se a reprodução sexuada. Nessa etapa ocorrem a recombinação e a mutação.

A recombinação (*crossover*) é o operador em permite a troca de material genético entre 2 pais. No caso mais comum, é iniciado sendo determinada uma posição aleatória, chamada de ponto de recombinação. A seguir tem-se o processo de mutação, que ocorre mediante uma probabilidade pré-determinada.

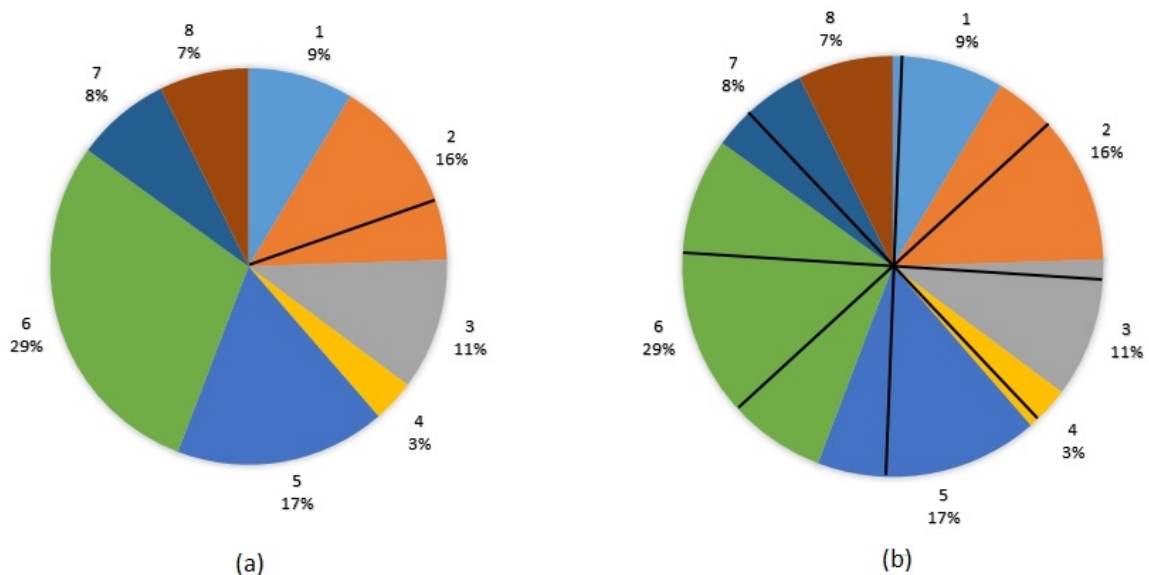
Após essas etapas, ocorre uma nova avaliação para obtenção o valor do *fitness* dos novos indivíduos gerados e, em seguida, verifica-se se foi atingido o critério de parada estabelecido. Caso contrário, retorna-se ao ciclo do algoritmo (seleção, recombinação e mutação). A cada nova geração o valor do *fitness* tende a ser maior em relação a geração anterior (GRIFFITHS et al., 2006).

3.1 SELEÇÃO

A seleção é o primeiro mecanismo de operação do Algoritmo Genético. Existem diversos tipos de métodos de seleção, sendo a roleta e o torneio os mais usuais (CASTRO, 2006).

No caso da roleta (Figura 5), é criada uma roleta viciada, em que as "fatias" serão proporcionais ao *fitness* de cada indivíduo. Logo, quanto melhor ele for, maior será a probabilidade de ser selecionado. Como desvantagem, a tendência é perder a diversidade rapidamente entre as gerações subsequentes. Ocorrerá uma quantidade de seleções realizadas dependendo de quantos indivíduos são necessários para atender a próxima etapa (MICHALEWICZ; HARTLEY, 1996).

Figura 5 – Roleta clássica (a), *Universal Stochastic Sampling* (b)



Fonte: Autoria própria.

Cada indivíduo recebe um valor de aptidão, sendo aqueles que para os estão mais adaptados são atribuídos um maior valor de *fitness*. Este valor de aptidão é o resultado da função de avaliação. A seguir, calcula-se um valor percentual em relação ao valor total das avaliações. Como exemplifica a Tabela 1, para o indivíduo 5 tem-se $85/500 = 17\%$ de chances de ser escolhido (LINDEN, 2008).

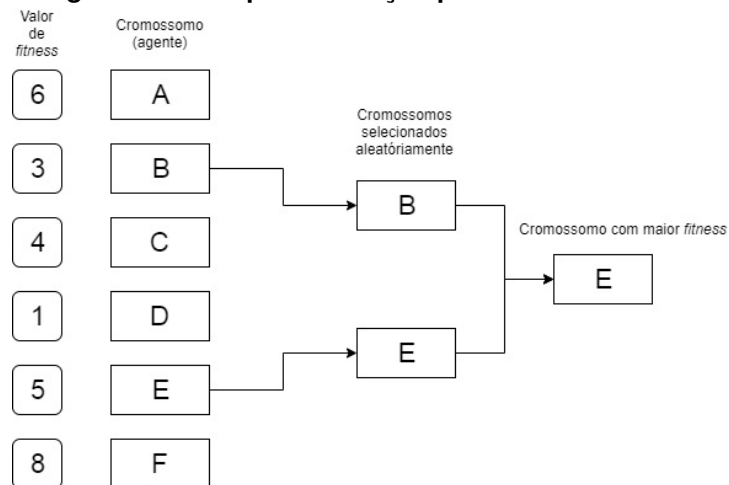
A variação da roleta chamada de Amostragem Estocástica Universal (*Stochastic Universal Sampling*) na Figura 5 (b), permite manter a variedade genética da população. O sorteio ocorre apenas uma vez, é definido uma distância igual entre cada indivíduo selecionado. Dessa forma, pode-se evitar de perder o indivíduo com maior

Tabela 1 – Exemplo de avaliação para a roleta

Indivíduo	Avaliação individual	Avaliação acumulada	Fitness individual %	Fitness acumulado %
1	45	45	9	9
2	80	125	16	24
3	55	180	11	35
4	15	195	3	39
5	85	280	17	56
6	145	425	29	85
7	40	465	8	93
8	35	500	7	100

Fonte: Autoria própria.

valor de *fitness* e ampliar as chances de os piores serem escolhidos, preservando a diversidade de genes (BÄCK; FOGEL; MICHALEWICZ, 2018).

Figura 6 – Exemplo de seleção por torneio

Fonte: Autoria própria.

No torneio, é sorteada uma quantidade pré-determinada de indivíduos da população para participar. Pela literatura essa quantidade comumente é de dois indivíduos. Em seguida, é feita a comparação do *fitness*, sendo o vencedor aquele que possui o maior valor, como pode-se observar na Figura 6. O torneio tem variações: a) os indivíduos perdedores voltam para a população e podem participar dos próximos torneios (caso seja sorteado); b) os indivíduos perdedores são eliminados. Este último é chamado de torneio da morte. Um ponto negativo desta variação é a redução da variabilidade genética, pois está eliminando possíveis soluções (MICHALEWICZ; HARTLEY, 1996).

Outro ponto sobre os métodos de seleção é que eles permitem perder o melhor indivíduo. Por ser gerado um valor aleatório para definir qual irá participar da próxima etapa, existe a probabilidade deste não ser escolhido.

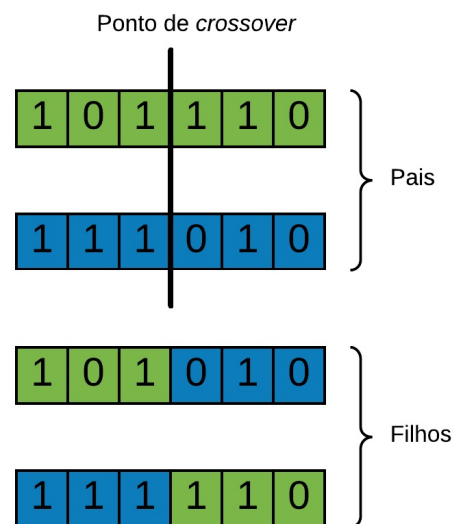
No geral, é sabido na literatura que o torneio tende a ser menos tendencioso que a roleta, imprimindo uma menor pressão seletiva na população. Isto é particularmente importante, pois a diversidade é um fator primordial para que o algoritmo não fique preso em mínimos locais ou apresente convergência prematura (CASTRO, 2006).

3.2 RECOMBINAÇÃO

A recombinação é o operador genético que permite que haja o cruzamento entre indivíduos selecionados, como ilustra a Figura 7. A partir deste operador será trocado o material genético, gerando novos indivíduos, os filhos ou prole (BÄCK; FOGEL; MICHALEWICZ, 2018).

Para a recombinação tanto na codificação binária quanto na codificação real existe a recombinação de um ponto, de múltiplos pontos, binário simulado, aritmética, etc (FILHO, 2018). Na Figura 7, exemplifica-se a recombinação de um ponto, em que a partir deste, ocorre a troca de genes.

Figura 7 – Recombinação binária para o AG.



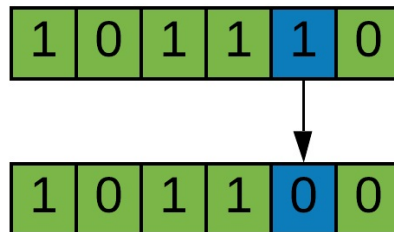
Fonte: Autoria própria.

A recombinação é o operador que permite o refinamento das soluções já encontradas (exploração), sendo um fator de convergência do AG.

3.3 MUTAÇÃO

Em cada gene do indivíduo, existe a probabilidade de ocorrer uma mutação, alterando o seu valor ou alelo, como exemplifica a Figura 8.

Figura 8 – Mutação binária.



Fonte: Autoria própria.

Comumente, essa troca é espúria, simulando o processo natural, e podendo ocorrer em qualquer da população. A taxa de mutação pode ser fixa, como ocorre na versão clássica, ou do tipo mutação variável. Para os casos em que o valor do *fitness* está estagnado por mais de 20 gerações, usa-se a mutação com valor elevado como meio de gerar diversidade. Por exemplo, pode-se admitir que essa taxa seja complementar ao valor do *fitness*, ou seja, se o melhor indivíduo apresenta valor de *fitness* de 0,6 (60% do valor unitário), a taxa de mutação será 40%.

A mutação é utilizada para permitir que a população escape de mínimos locais e que leve indivíduos, aleatoriamente, à regiões e bacias de atração ainda inexploradas pelo método. Permite assim que o AG faça buscas globais (exploração), sendo um fator de divergência.

3.4 VARIANTES DO ALGORITMO GENÉTICO

Para cada etapa do processo de implementação de um AG, existem variações nos tipos de seleção, recombinação e mutação. No AG clássico os filhos assumem o lugar de seus pais, mas também pode-se acrescentar populações intermediárias, na qual os filhos e os pais estão contidos e concorrem para a próxima geração (HOLLAND et al., 1992).

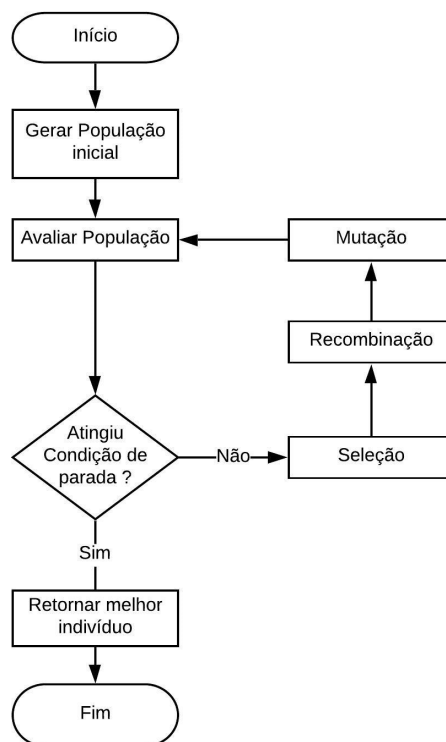
Para evitar a perda dos melhores indivíduos, pode-se usar o elitismo, no qual os indivíduos são agregados para melhorar os métodos de seleção. Os melhores da-

quela geração são escolhidos para comporem a população da geração seguinte, embora possa levar a uma perda rápida de diversidade.

Todas essas variações nas etapas foram elaboradas ao longo do tempo com o propósito de melhorar o desempenho do algoritmo (BÄCK; FOGEL; MICHALEWICZ, 2018).

As etapas de implementação e execução do AG são sumarizadas na Figura 9 e no Algoritmo 1.

Figura 9 – Fluxograma Algoritmo Genético.



Fonte: Autoria própria.

Algoritmo 1 – Algoritmo Genético.

- 1: **Início**
 - 2: $i = 0$
 - 3: Gerar população inicial $P(i)$
 - 4: Definir o número máximo de iterações MI
 - 5: **para** $i : MI$ **faça**
 - 6: Avaliar população $P(i)$
 - 7: Selecionar pares da população $P(i)$
 - 8: Realizar recombinação
 - 9: Realizar mutação
 - 10: **finaliza para**
 - 11: Retornar melhor indivíduo
 - 12: **Fim**
-

Fonte: Autoria própria.

Neste trabalho foram consideradas 10 versões do Algoritmo Genético:

- AG1: Clássico, seleção por roleta, com 70% de probabilidade recombinação, sendo este de um ponto e mutação fixa;
- AG2: Seleção por roleta, recombinação de ponto e mutação fixa;
- AG3: Seleção por roleta com subpopulação, recombinação de ponto e mutação fixa;
- AG4: Seleção por torneio, recombinação de ponto e mutação fixa;
- AG5: Seleção por torneio com subpopulação, recombinação de ponto e mutação fixa;
- AG6: Seleção por torneio, recombinação de ponto e taxa de mutação em 20%.
- AG7: Seleção por torneio da morte, recombinação de ponto e mutação fixa;
- AG8: Seleção por roleta *Stochastic Universal Sampling*, recombinação de ponto e mutação fixa;
- AG9: Seleção por torneio, recombinação de ponto e mutação variável;
- AG10: Seleção por torneio da morte, recombinação de ponto e mutação variável;

Considerar a taxa de mutação de 10% e recombinação de 100% quando não mencionadas.

4 EVOLUÇÃO DIFERENCIAL

A Evolução Diferencial (ED), proposta por Rainer Storn e Kenneth Price em 1995, foi inicialmente criada para solucionar problemas de otimização contínua de forma robusta. É um método populacional na qual as possíveis soluções são obtidas através das combinações entre os vetores (agentes) da população. A ED é um algoritmo evolutivo que se utiliza de mecanismos semelhantes ao Algoritmo Genético: seleção, mutação e recombinação (PRICE; STORN; LAMPINEN, 2006).

A ED é inicializada com uma população aleatória de vetores. Dentre estes, um é escolhido de forma arbitrária para ser utilizado nos processos de combinações, sendo denominado *Target Vector* ($\mathbf{X}_{(r1,G)}$). O processo de mutação ocorre com outros dois vetores escolhidos de maneira aleatória ($\mathbf{X}_{(r2,G)}$ e $\mathbf{X}_{(r3,G)}$), com os quais é obtida uma diferença ponderada que será adicionada ao *Target Vector* resultando no vetor mutado ($\mathbf{V}_{ji,G}$).

Na sequência ocorre a recombinação vetorial entre o vetor mutado e o *Target Vector*, tendo como produto o *Trial Vector* ($\mathbf{U}_{ji,G}$). Esta é a etapa que chama-se de recombinação. Por fim, é feita a avaliação de *fitness* do *Trial Vector*, depois sendo este comparado ao *Target Vector*. O agente que possuir o melhor *fitness* segue para a próxima geração. Tal etapa equivale à seleção. Esse procedimento se repete de forma que cada um dos vetores da atual população seja usado uma vez como *Target Vector* (CHATTOPADHYAY; MISHRA; GOSWAMI, 2016).

Neste estudo, são abordadas duas propostas para a ED, uma binária e outra booleana. As diferenças desses algoritmos encontram-se apenas na etapa de mutação. A booleana utiliza operadores lógicos para realizar seus cálculos, enquanto que a versão binária usa operações aritméticas e uma função de transferência.

A mutação se inicia escolhendo aleatoriamente os vetores (agentes) para participar do processo. No caso booleano, o vetor mutado é gerado de acordo com a Equação 1 (ZHANG et al., 2010):

$$\mathbf{V}_{(ji,G)}^{\text{bo}} = \mathbf{X}_{(r1,G)} + \mathbf{F} \bullet (\mathbf{X}_{(r3,G)} \oplus \mathbf{X}_{(r2,G)}) \quad (1)$$

em que F é outro vetor gerado randomicamente, G é a geração atual, $r1, r2$ e $r3$ são os índices de vetores aleatoriamente escolhidos e os operadores lógicos são OR (+),

AND (\bullet) e XOR (\oplus)

Para a mutação binária utiliza-se de operações aritméticas como soma, subtração e multiplicação, como mostra a Equação 2:

$$\mathbf{V}_{(ji,G)}^{\text{bn}} = \mathbf{X}_{(r1,G)} + F(\mathbf{X}_{(r3,G)} - \mathbf{X}_{(r2,G)}) \quad (2)$$

O peso F é uma variável real, sendo $F = 0,5$ ou $0,8$ valores encontrados comumente na literatura (DENG et al., 2010). Esse parâmetro é utilizado para reduzir a pressão seletiva. Considerando $F < 1$ obtém-se soluções mais diversas, porém se F for muito pequeno, valores menores que $0,4$ impactam rapidamente na redução da diversidade induzindo à uma convergência prematura. Dessa maneira, o valor atribuído deve estar no intervalo de $0,4 < F < 1$, mantendo a variedade da população no decorrer das gerações (PRICE; STORN; LAMPINEN, 2006).

Ao final das operações, tem-se um vetor mutado $\mathbf{V}_{(ij,G)}^{\text{bn}}$ com suas componentes em valores reais. Logo, é necessário um modo para converter esse vetor para o domínio binário. O recurso aplicado para a conversão é através de uma função de transferência.

A Sigmoide (*S-Shaped*) é a função de transferência escolhida neste trabalho, por ser a mais utilizada na literatura. As curvas em forma de S são aplicadas em pesquisas de dinâmica populacional e análises econômicas, devido aos estudos da lei do crescimento natural. Sua aplicação se deve aos bons resultados obtidos em estudos de extrapolação de tendências com aeronaves, microeletrônica, sistemas de conversão de energia, aceleradores de partículas e etc. Na medicina é usada para descrever a difusão de doenças epidêmicas e na biologia é utilizada para retratar o crescimento de uma espécie em competição (KUCHARAVY; GUIO, 2011). A Sigmoide pertence ao grupo das curvas em S e devido aos seus resultados na literatura, é a função definida para esta etapa.

É realizada, neste caso, a conversão de cada dimensão conforme Equação 3:

$$\mathbf{S}(\mathbf{V}_{ji,D}) = \frac{1}{1 + e^{-V_{ji,D}}} \quad (3)$$

Na sequência o vetor binário é obtido pela Equação 4:

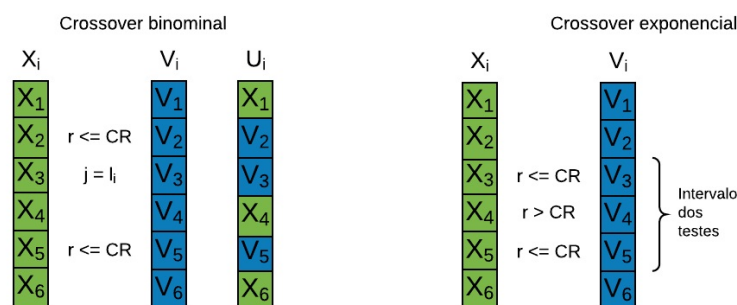
$$V_{ji,D}^{\text{bin}} = \begin{cases} 1 & , \text{ se } \text{rand}(0,1) \leq S(V_{ij,D}) \\ 0 & , \text{ caso contrário.} \end{cases} \quad (4)$$

Para o *rand* é sorteado um valor pertencente ao intervalo entre 0 e 1. O índice i é cada indivíduo e j é cada componente do mesmo. Tendo o vetor mutado com todos seus valores em binário, pode-se dar continuidade para a próxima etapa. O *Trial Vector* é o resultado da recombinação entre o vetor mutado e o *Target Vector*. Para isto, é realizado um teste condicional entre r_j e uma nova variável CR , com valor escolhido pelo usuário, seguindo o critério da Equação 5:

$$U_{ji,G} = \begin{cases} V_{ji,G} & , \text{ se } r_j \leq CR \text{ ou } j = l_i. \\ X_{ji,G} & , \text{ se } r_j > CR \text{ ou } j \neq l_i. \end{cases} \quad (5)$$

Esse critério segue para a recombinação binomial, na qual todos os componentes do vetor participam do teste condicional (Figura 10). Para esse caso, r_j é um valor randômico entre 0 e 1. A fim de garantir que o *Trial Vector* ($U_{ji,G}$) receba pelo menos um componente de $V_{ji,G}$, tem-se o índice l_i escolhido arbitrariamente entre 1 e D , e temos CR , que varia entre 0 e 1.

Figura 10 – Recombinação da Evolução Diferencial.



Fonte: Autoria própria.

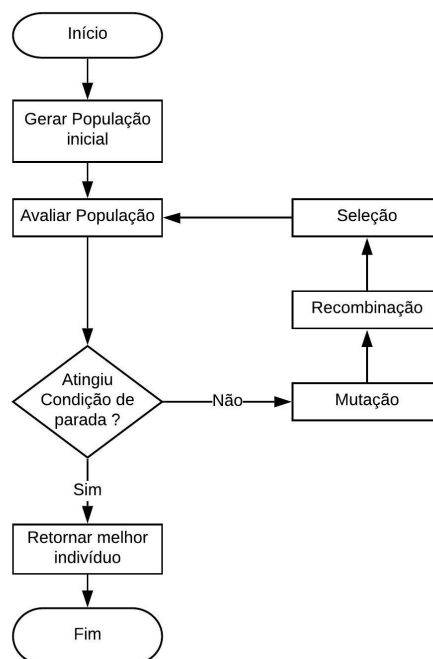
Na recombinação (*crossover*) exponencial se observa algumas diferenças em relação ao binomial conforme descreve (OPARA; ARABAS, 2019). Neste caso, o teste condicional não é feito para todos os componentes dos vetores, como ilustra a Figura 10. Antes é determinado aleatoriamente um intervalo que irá indicar o ponto de

início e final para o teste. O *Trial Vector* irá receber do *Target Vector* os componentes que não participaram do processo.

Na seleção é realizada a comparação dos valores de *fitness* do *Target* e *Trial Vector*. O agente com o melhor *fitness* é escolhido para compor a próxima geração.

Esse processo irá se repetir até a condição de parada ser atendida, como atingir o número máximo de iterações definido. Na Figura 11 pode-se observar todo o processo e no Algoritmo 2 se tem o pseudocódigo.

Figura 11 – Fluxograma Evolução Diferencial.



Fonte: Autoria própria.

Algoritmo 2 – Evolução Diferencial.

- 1: **Início**
 - 2: $i = 0$
 - 3: Gerar população inicial $P(i)$
 - 4: Definir o número máximo de iterações MI
 - 5: **para** $i : MI$ **faça**
 - 6: Avaliar população $P(i)$
 - 7: Realizar mutação
 - 8: Realizar recombinação
 - 9: Realizar seleção $P(i)$
 - 10: **finaliza para**
 - 11: Retornar melhor indivíduo
 - 12: **Fim**
-

Fonte: Autoria própria.

Em relação aos algoritmos de Evolução Diferencial, este trabalho aborda 10 variações. Estas possuem uma nomenclatura padrão de classificação para apresentar de maneira simplificada qual estratégia o algoritmo utiliza. Para o tipo de mutação pode-se ter *Rand*, *Best* e *Rand – to – Best*. Na quantidade de diferenças ponderadas tem-se 1 ou 2. Por fim, os tipos de recombinação que podem ser usados são o Binomial ou Exponencial. A lista das possibilidades segue abaixo:

- ED1: Rand/1/Bin
- ED2: Best/1/Bin
- ED3: Rand/2/Bin
- ED4: Best/2/Bin
- ED5: Rand-to-Best/2/Bin
- ED6: Rand/1/Exp
- ED7: Best/1/Exp
- ED8: Rand/2/Exp
- ED9: Best/2/Exp
- ED10: Rand-to-Best/2/Exp

Se o algoritmo apresenta a classificação Rand/1/Exp, significa que possui a mutação tipo *Rand* com uma diferença ponderada e a recombinação do tipo exponencial.

Os algoritmos apresentados para a Evolução Diferencial Binária estão definidos do ED1 ao ED10 e para Evolução Diferencial Booleana do ED11 ao ED20.

5 FUNÇÃO BENCHMARK

No campo da otimização existe uma diversidade de técnicas para encontrar soluções de problemas práticos, como as metaheurísticas bio-inspiradas. Estas são estratégias candidatas a solucionar problemas reais e operam com as restrições que envolvem as variáveis naquela determinada função custo. Por meio de escolhas de base aleatórias e mecanismos e operadores inspirados na natureza, tais métodos realizam recombinações para se orientar no espaço de busca e em seus arredores, com a capacidade de escapar de mínimos locais e realizar busca global (COUTO; SILVA; BARSANTE, 2015).

O termo função Benchmark ou de teste, faz referência à problemas de natureza didática, que permitem a comparação de ferramentas e métodos computacionais, com intuito de avaliar o desempenho relativo de cada uma. Esses problemas podem ser reais ou funções teste sintéticas, como caixeiro viajante (SILVA et al., 2017), *job shop* (MAGALHAES-MENDES, 2013), *OneMax problem* (DU et al., 2014), Problema da Mochila (ANDRÉ; PARPINELLI, 2015) entre outros. Ao aplicar métodos de otimização a tais problemas, almeja-se encontrar, dependendo do objetivo, o máximo ou mínimo global (DIETERICH; HARTKE, 2012).

O *OneMax* é o problema mais simples de otimização binária visando a maximização. Esta função foi escolhidas por ser de fácil implementação e compreensão, assim como o problema da mochila, sendo um modo rápido e simples para testar a eficiência de algoritmos binários.

5.1 ONEMAX PROBLEM

Um dos problemas testes muito utilizado em otimização binária é o *OneMax Problem*, no qual busca-se maximizar o número de "1s" em uma população. Inicia-se com uma quantidade pré-determinada de indivíduos, sendo cada um representado por um vetor que possui uma quantidade mista de bits 0 e 1. A meta é encontrar o agente com a maior quantidade de "1", tendo como melhor resultado esperado um vetor com todos os bits iguais a "1" (DU et al., 2014).

Uma possível função para a avaliação do *fitness* é feita através da Equação 6:

$$Fitness = \frac{\sum_{d=1}^D i(d)}{D} \quad (6)$$

em que D é o número de dimensões do vetor e $i(d)$ as dimensões que apresentam o valor 1.

Caso tenha um problema com 10 dimensões, tem-se a Tabela 2 como exemplo:

Tabela 2 – Cálculo de *fitness* para *OneMax Problem*

Indivíduo	<i>Fitness</i>
0010011101	0,5
1010011111	0,7
0010001001	0,3
1110111111	0,9

Fonte: Autoria própria.

Para este trabalho foram definidas 4 valores de dimensões para as avaliações de desempenho: 50, 100, 250 e 500.

5.2 PROBLEMA DA MOCHILA

O Problema da Mochila (*Knapsack Problem*) é um problema de otimização de natureza combinatória, na qual seu objetivo é maximizar o valor dos itens em um mochila sem ultrapassar sua capacidade. Tendo um grupo de itens apresentados, o algoritmo deve encontrar a alocação ideal para a mochila. Cada item é caracterizado por seu volume e valor de lucro (quanto "vale a peça"), e a mochila possui capacidade máxima. A melhor combinação é aquela em que o peso total seja menor ou igual à capacidade da mochila e o valor seja máximo entre as combinações encontradas (SINGH, 2011).

Existem variações para o problema como a mochila inteira, em que o mesmo item pode se repetir dentro da mochila; mochila compartimentada, na qual os itens são separados por classe e cada compartimento somente pode ter itens de mesma classe; Múltiplas mochilas, na qual se pretende preencher várias delas; e a Mochila 0/1 ou também conhecida por mochila binária, que não permite repetição e apenas indica se o item está ou não presente (COTTA; TROYA, 1998). Nesse estudo, escolheu-se a mochila binária para realizar o testes nos algoritmos.

Tabela 3 – Tabela de itens com seus respectivos valores e pesos

<i>Item(X)</i>	1	2	3
<i>Valor(V)</i>	6	5	7
<i>Peso(P)</i>	3	4	5

Fonte: Autoria própria.

Considere um problema que apresenta 3 itens como na Tabela 3:

Havendo 3 itens, temos $2^3 = 8$ possíveis combinações para a mochila. Se considerarmos a sua capacidade igual a 10 (peso máximo suportado), busca-se a solução mais adequada para este caso. Os itens que estarão inclusos na mochila tem em suas posições o bit 1 e, se a capacidade da mochila for ultrapassada, o resultado do valor para a combinação é desconsiderado.

Para a resolução deste problema, tem-se a quantidade de colunas igual a capacidade da mochila e o número de linhas será a quantidade de itens. A solução é o valor da última linha e coluna:

Tabela 4 – Resolução para o problema da mochila

	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	6	6	6	6	6	6	6	6
2	0	0	0	6	6	6	6	11	11	11	11
3	0	0	0	6	6	7	7	11	13	13	13

Fonte: Autoria própria.

$$R(i,C) = \begin{cases} R(i-1,C) & , \text{ se } P_i > C \\ \max (R(i,C), R(i-1,C - P_i) + V_i) & , \text{ se } P_i \leq C \end{cases} \quad (7)$$

Tem-se $R(i,C)$ como o máximo valor encontrado combinando os pesos P_i dos itens, sendo $1 \leq i \leq 3$ e C representando a capacidade atual da mochila que se encontra no intervalo de $0 \leq C \leq 10$. Para obter-se cada valor de $R(i,C)$, inicia-se uma varredura com a capacidade atual da mochila (colunas) utilizando cada item (linha), verificando-se se o peso do item é suportado pela mochila. Se verdadeiro, é atribuído o valor do item na posição $R(i,C)$. Para cada linha na varredura são considerados os itens das linhas anteriores e o item da linha atual. Esse processo ocorre para cada linha na Tabela 4. Caso a presença do novo item leve a extrapolação da capacidade, considera-se o valor anterior. Se a capacidade não for ultrapassada é atribuído o valor máximo de acordo com a Equação 7 (YANASSE; SOMA, 1987).

A melhor solução encontrada contém os itens 1 e 3 obtendo peso igual a 8 e o valor igual a 13, estando de acordo com capacidade da mochila e por ter atingido o máximo valor entre todas as combinações.

Com dimensões reduzidas outras técnicas conseguem alcançar os ótimos globais, porém conforme estas forem aumentando, destaca-se a potencialidade de busca de métodos de otimização para obter-se um melhor resultado e de maneira mais rápida.

Para o Problema da Mochila os valores das combinações são considerados como o *fitness*. Neste trabalho, as simulações foram realizadas com dimensões variadas, sendo a quantidade de itens que define o número de dimensões de um indivíduo. Na Tabela 5, pode-se verificar a capacidade da mochila de acordo com as dimensões abordadas para este trabalho.

Tabela 5 – Características para cada variação da mochila

Quantidade de itens	Capacidade da mochila
30	629
100	1159
250	719
500	3422

Fonte: Autoria própria.

6 RESULTADOS E DISCUSSÃO

Neste Capítulo serão discutidos os resultados computacionais obtidos pelo Algoritmo Genético e a Evolução Diferencial para otimização binária. Todas as variantes descritas nos Capítulos 3 e 4 têm como condições de parada 1000 gerações. Ainda, para o *OneMax Problem* se aplica a condição do *fitness* ser igual a 1.

Afim de se apresentar uma métrica para comparar os desempenhos, armazenou-se a melhor execução, ou seja, a que encontrou o valor mais alto de *fitness*. Foram realizadas 100 simulações independentes para cada versão de algoritmo e calculada uma média dos valores obtidos. Também se apresentam os gráficos tipo boxplot. O boxplot é um gráfico estatístico que apresenta a mediana, o primeiro e terceiro quartil, o maior e menor valor do grupo de dados e é possível visualizar a dispersão dos resultados encontrados.

Todas as simulações foram feitas utilizando o software MATLAB®. Na sequência, apresenta-se os resultados obtidos separados por função com diferentes números de dimensões.

6.1 FUNÇÃO ONEMAX PROBLEM

Otimizou-se o *OneMax Problem* com uma população de 20 indivíduos para cada combinação do AG e da ED. Os resultados das simulações de todas as dimensões ($D=50, 100, 250$ e 500) podem ser verificados na Tabela 6. Os respectivos gráficos boxplot dos valores de *fitness* podem ser observados nas Figuras 12, 13, 14 e 15.

A média de iterações para todas as dimensões ficou igual a 1000 ou muito próximas desse valor.

Tabela 6 – Média de convergência do *fitness* para cada dimensão do *OneMax Problem*

Algoritmo	50	100	250	500
AG1	0,76	0,65	0,59	0,56
AG2	0,76	0,65	0,59	0,56
AG3	0,96	0,92	0,85	0,76
AG4	0,96	0,92	0,84	0,77
AG5	0,96	0,92	0,84	0,77
AG6	0,96	0,92	0,84	0,76
AG7	0,93	0,86	0,76	0,68
AG8	0,99	0,98	0,91	0,81
AG9	0,96	0,92	0,85	0,77
AG10	0,93	0,87	0,76	0,68
ED1	1	1	1	1
ED2	1	1	1	1
ED3	1	1	1	1
ED4	1	1	1	1
ED5	1	1	1	1
ED6	0,95	0,77	0,63	0,58
ED7	0,92	0,75	0,63	0,58
ED8	0,95	0,76	0,63	0,58
ED9	0,92	0,75	0,63	0,58
ED10	0,92	0,75	0,63	0,58
ED11	0,83	0,80	0,77	0,76
ED12	0,81	0,78	0,75	0,73
ED13	0,82	0,80	0,77	0,76
ED14	0,82	0,79	0,78	0,77
ED15	0,81	0,77	0,75	0,72
ED16	0,78	0,68	0,59	0,56
ED17	0,69	0,62	0,57	0,55
ED18	0,79	0,67	0,59	0,56
ED19	0,72	0,64	0,58	0,55
ED20	0,68	0,62	0,57	0,55

Fonte: Autoria própria.

Os resultados computacionais relativos ao *OneMax Problem* para todos os cenários propostos (4 variações no número de dimensões) foram semelhantes ao uso do Algoritmo Genético. De forma geral, o método de seleção por torneio levou a uma melhor performance aliada a uma menor dispersão em relação ao uso da roleta (AG1 e AG2). Os AG3 e AG5 usaram o recurso da subpopulação, na qual os pais e filhos competem para compor a próxima geração. Embora o AG3 também tenha usado a seleção do tipo roleta, o efeito negativo desse método de seleção foi compensado, já que esta proposta apresentou um desempenho semelhante ao torneio. Todavia, na subpopulação com o torneio não se percebeu melhoria efetiva, já que a proposta chegou a resultados semelhante aos que não a utilizaram como o AG6. É importante salientar que esse estratagema incrementa o custo computacional.

Para o Algoritmo Genético nas 4 dimensões aplicadas, o AG8 teve melhor performance com suas médias de *fitness*.

Dentre a estratégia booleana para 50 dimensões o ED11 obteve desempenho próximo aos ED12, ED13, ED14 e ED15 e resultado melhor que o ED19 e ED20. Além disso, apresentaram maior dispersão os ED12 e ED15. Em 100 dimensões, o ED11 e ED13 mostraram reposta superior aos outros, até mesmo sobre os ED6 a ED10. Com 250 e 500 dimensões o ED14 alcançou valores melhores que os ED16 ao ED20.

Para as versões booleanas no *OneMax Problem*, as estratégias com 2 diferenças ponderadas (ED13 e ED14) se destacaram, tendo médias melhores. Para 50 e 100 dimensões são semelhantes e com 250 e 500 ficam mais evidentes. A recombinação exponencial (ED16 a ED20) levou a resultados inferiores em relação às estratégias que fazem uso do binomial (ED11 ao ED15).

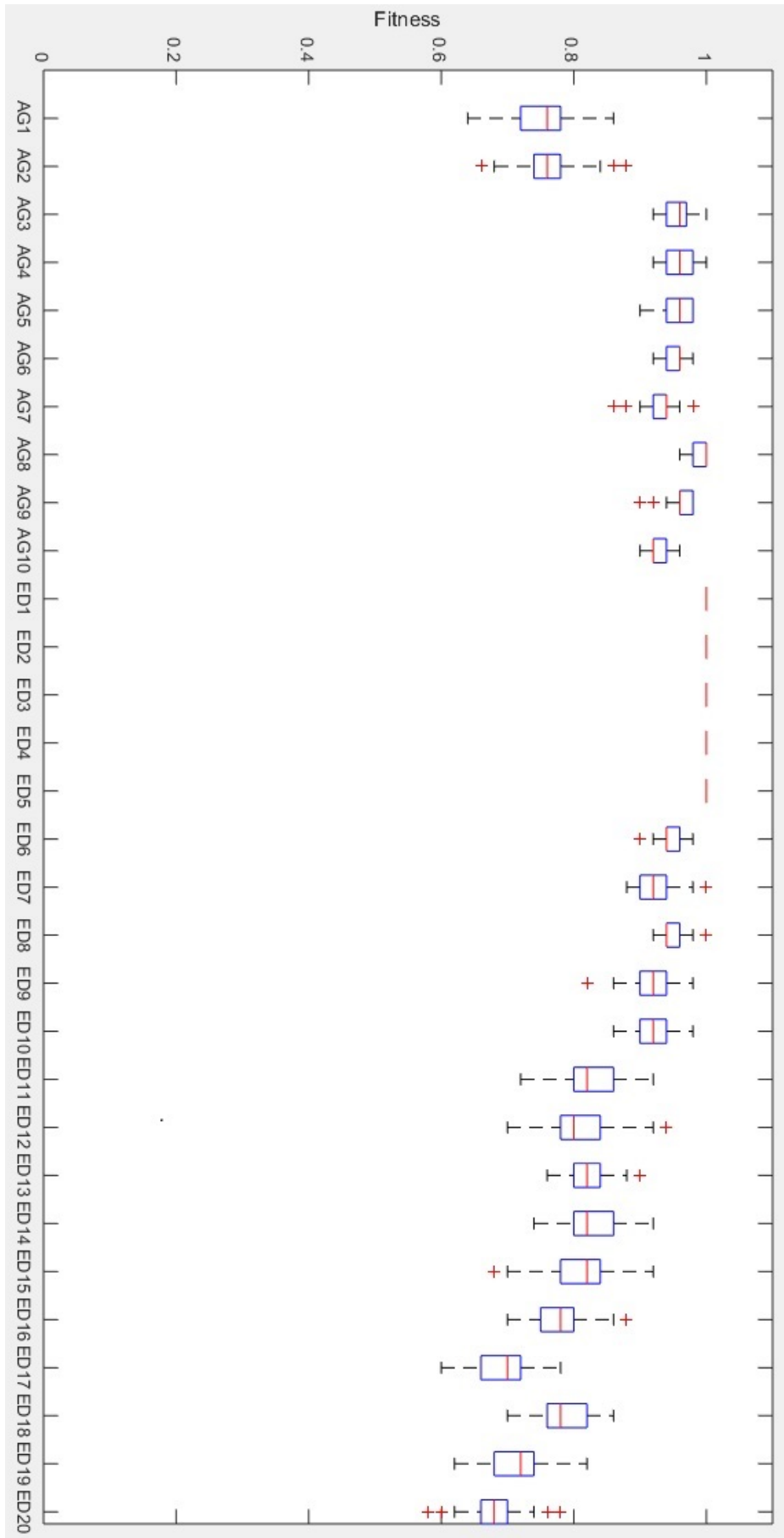
Para todas as dimensões do *OneMax*, os ED binários, a ED1 até a ED5 se destacaram por seu desempenho superior, chegando a resultar em valores de *fitness* unitário. Estes utilizaram a recombinação binomial, enquanto os demais a recombinação exponencial (ED6 a ED10).

Com 50 dimensões, a recombinação exponencial do ED binário (ED 6 ao ED10) obteve valores de convergência maiores que o ED booleano com recombinação binomial (ED11 ao ED15). Considerando 100 dimensões, os valores foram semelhantes e, para 250 e 500 a situação se inverte, tendo (ED11 ao ED15) com resultados superiores.

Entre as propostas de Evolução Diferencial os melhores desempenhos utilizaram a recombinação binomial, sendo ED1 ao ED5 e ED11 ao ED15, conforme a Figura 15 e a versão binária teve performance melhor que a booleana.

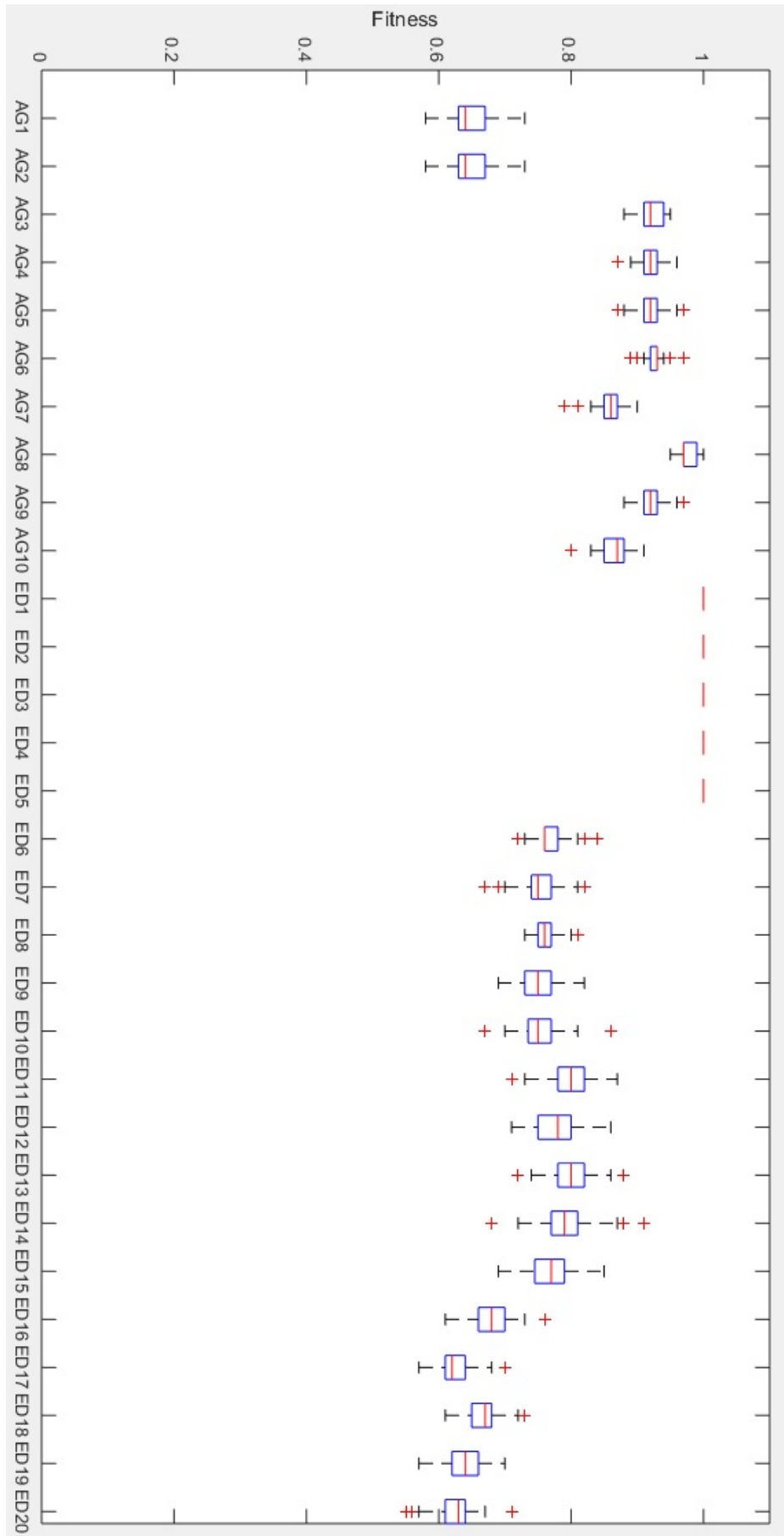
Nota-se um grau menor de dispersão conforme aumentou-se o valor da dimensão, comparando a Figura 14 com a Figura 15.

Figura 12 – Boxplot do Onemax Problem para 50 dimensões.



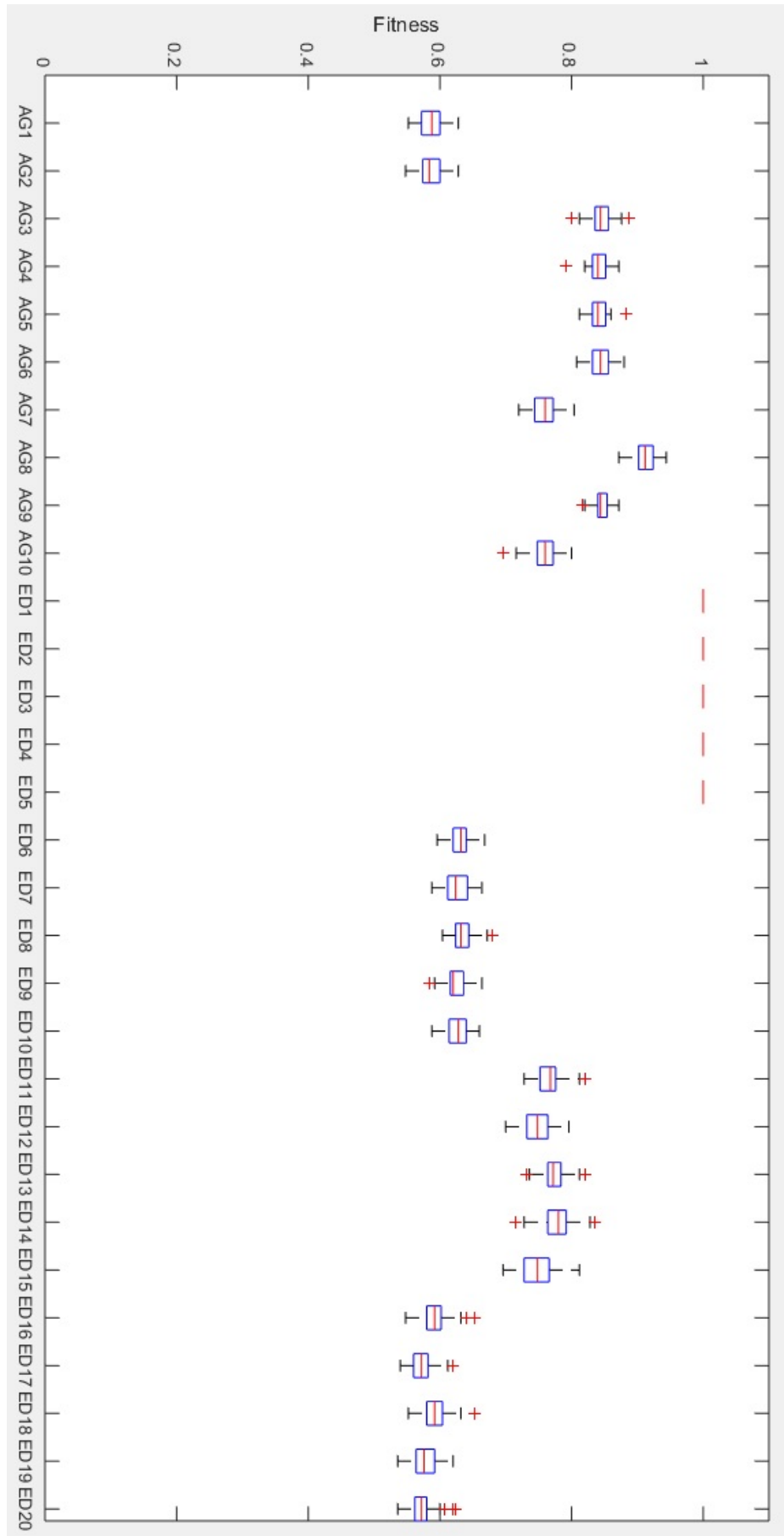
Fonte: autoria própria.

Figura 13 – Boxplot do Onemax Problem para 100 dimensões.



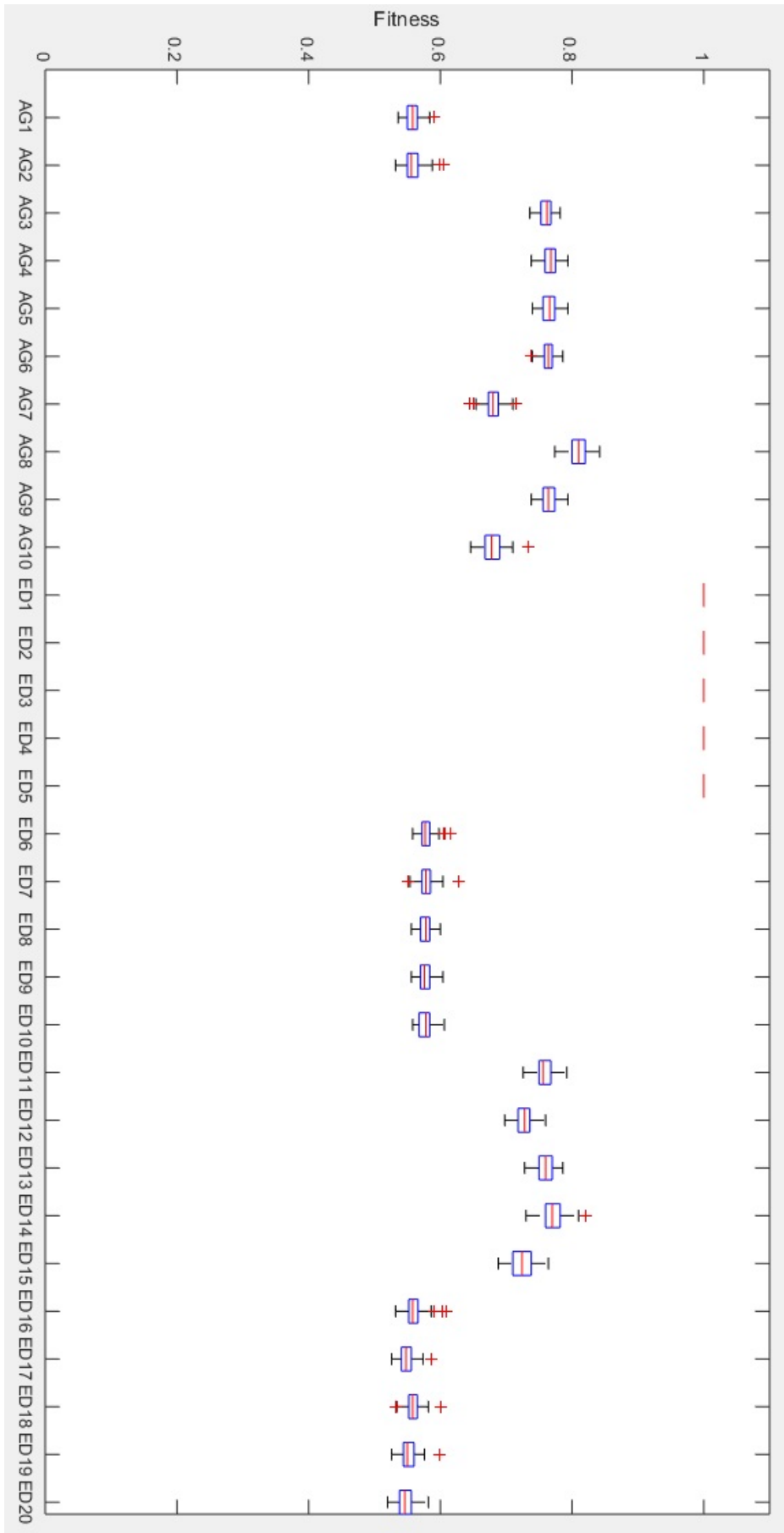
Fonte: autoria própria.

Figura 14 – Boxplot do Onemax Problem para 250 dimensões.



Fonte: autoria própria.

Figura 15 – Boxplot do Onemax Problem para 500 dimensões.



Fonte: autoria própria.

6.2 PROBLEMA DA MOCHILA

O Problema da Mochila binária foi otimizado tendo 20 indivíduos contidos em sua população para cada combinação do AG e da ED. O resultado das simulações contendo 30, 100, 250 e 500 itens (dimensões), e a média dos parâmetros obtidos para a função podem ser analisados nas Tabelas 7, 8, 9 e 10 ressaltando as melhores estratégia de AG e ED em negrito. Os gráficos boxplot dos valores de *fitness* das simulações podem ser vistos nas Figuras 16, 17, 18 e 19.

6.2.1 30 dimensões para problema da mochila

Para 30 itens a capacidade da mochila é 629.

Tabela 7 – Média de convergência para 30 dimensões do problema da mochila

Algoritmo	Valor	Peso	Iteração
AG1	684,18	623,07	82,39
AG2	682,16	622,38	141,82
AG3	672,59	619,48	1,06
AG4	676,74	621,69	17,31
AG5	673,56	620,58	1,08
AG6	675,69	621,13	46,67
AG7	699,62	625,78	526,6
AG8	681,61	622,77	66,92
AG9	676,27	619,71	1,27
AG10	674,45	620,24	1,28
ED1	693,51	624,37	352,99
ED2	701,68	625,46	462,41
ED3	693,33	623,19	328,81
ED4	705,15	625,89	437,28
ED5	703,59	625,39	425,74
ED6	702,90	626,08	396,23
ED7	707,43	626,90	404,91
ED8	701,97	625,93	396,02
ED9	706,54	626,62	426,19
ED10	706,91	626,66	428,49
ED11	684,67	624,11	33,28
ED12	677,27	622,43	1,29
ED13	684,65	624,24	45,17
ED14	676,70	621,17	1,13
ED15	676,93	621,72	1,29
ED16	684,83	624,27	28,61
ED17	676,18	622,12	1,27
ED18	684,70	624,33	36,32
ED19	676,16	622,46	1,23
ED20	675,60	622,09	1,27

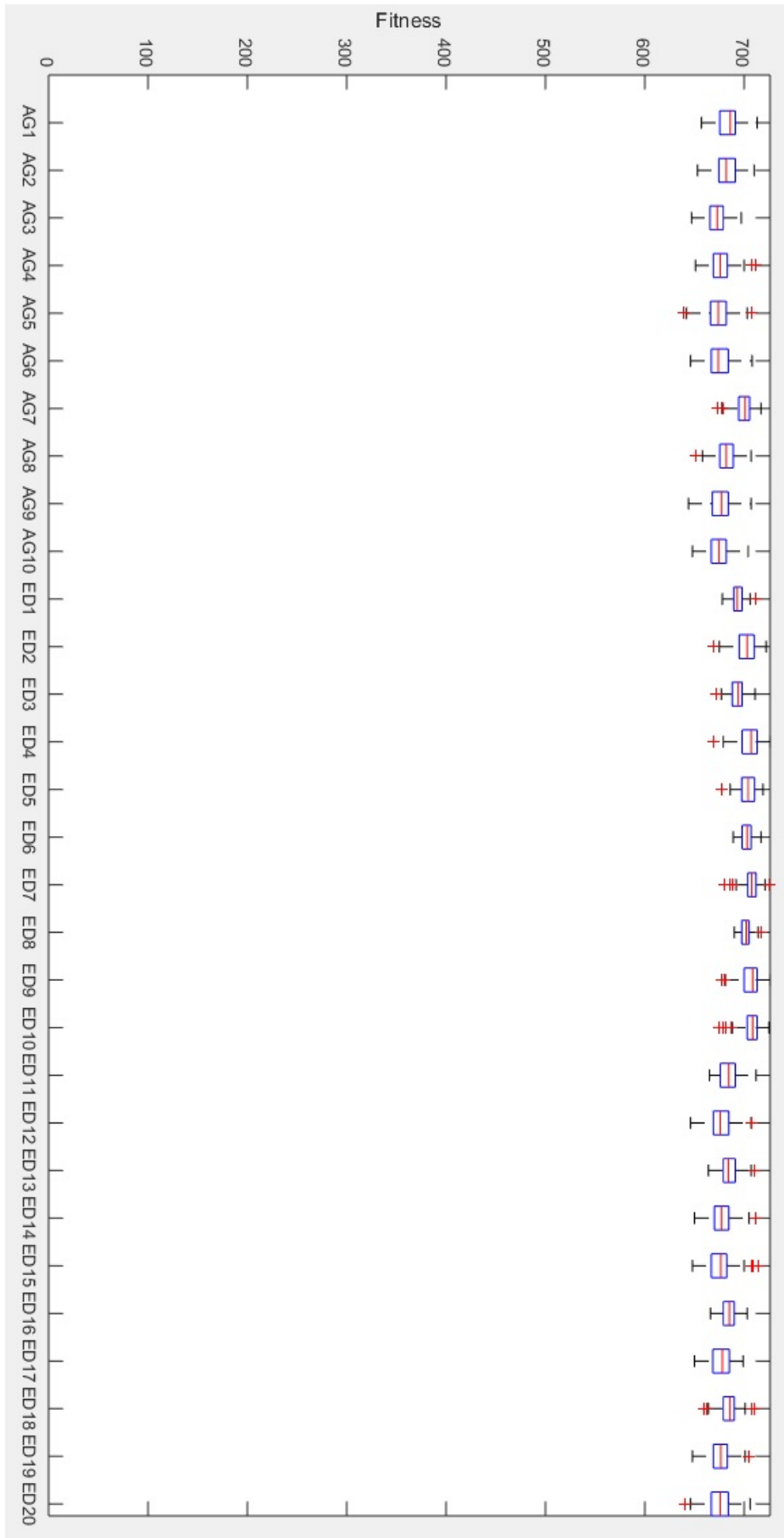
Fonte: Autoria própria.

As EDs de índice 1 a 10 são versões binárias, enquanto da ED11 a ED20 são

as booleanas. O AG7 entre os genéticos obteve melhores médias de *fitness*. O ED7 e ED 16 entre suas versões, obtiveram as médias mais altas. Observando a Figura 16, os resultados foram próximos para todos os algoritmos.

Na Tabela 7 observa-se que os algoritmos com maiores médias de iterações, também possuem maiores valores de *fitness*, ou seja, não levam a convergência prematura como ocorre por exemplo em alguns genéticos como AG3 e AG9 e a Evolução Diferencial booleana como os ED12, ED14, ED15, ED17, ED19 e ED20.

Figura 16 – Boxplot para mochila de 30 dimensões.



Fonte: autoria própria.

6.2.2 100 dimensões para problema da mochila

Para 100 itens a capacidade da mochila é 1159.

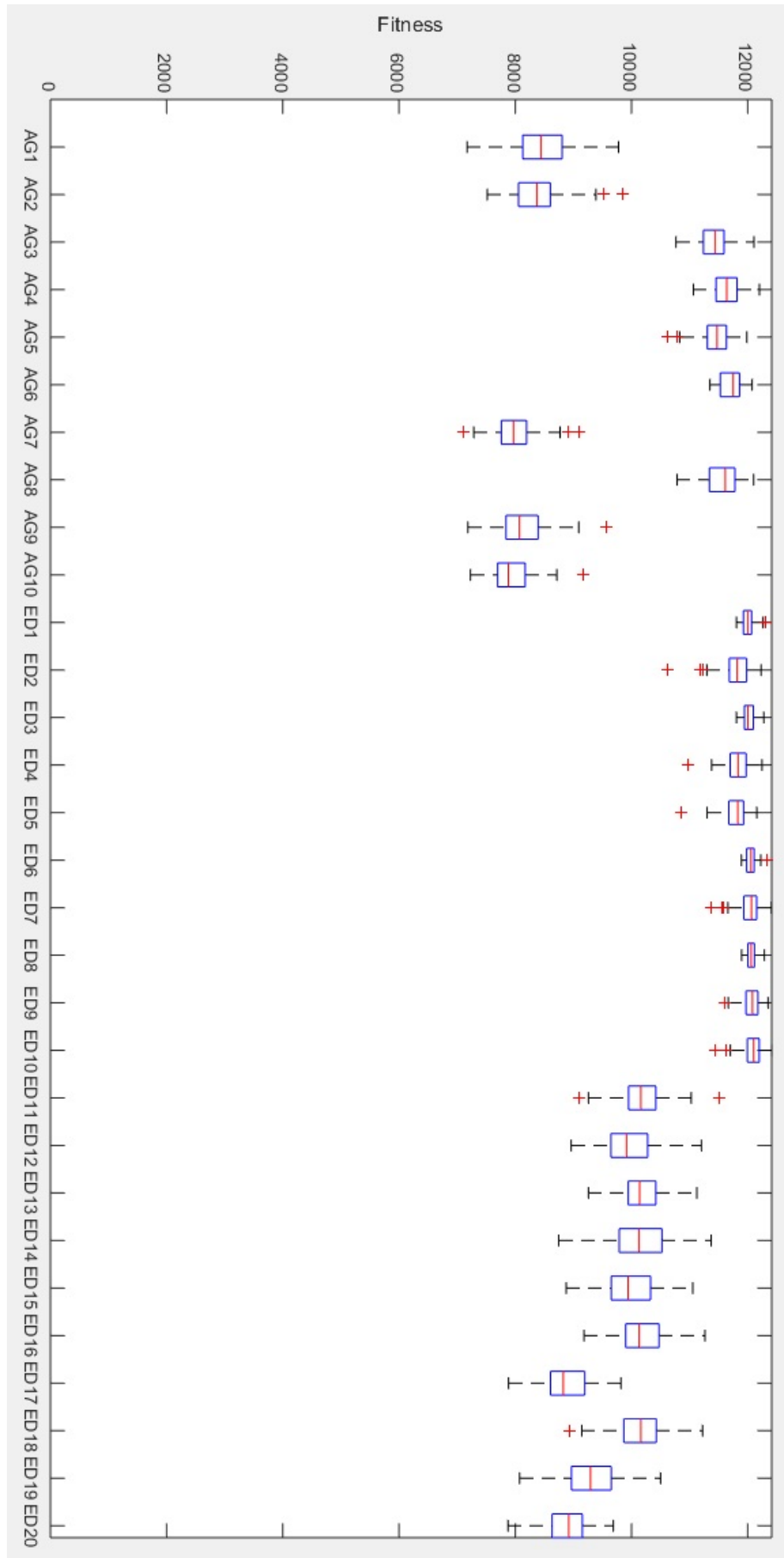
Tabela 8 – Média de convergência para 100 dimensões da mochila

Algoritmo	Valor	Peso	Iteração
AG1	8494,37	797,15	203,14
AG2	8392,87	787,19	202,64
AG3	11426,65	1151,48	664,33
AG4	11628,86	1156,81	633,9
AG5	11453,89	1151,66	642,29
AG6	11706,15	1156,61	517,49
AG7	7991,07	768,72	1,48
AG8	11565,16	1057,78	877,27
AG9	8106,11	763,67	2,85
AG10	7945,29	768,79	1,63
ED1	11999,78	1156,63	150,74
ED2	11812,81	1156,83	10,97
ED3	12017,61	1156,6	143,85
ED4	11816,06	1157,45	20,13
ED5	11795,85	1157,05	21,32
ED6	12052,35	1155,78	575,68
ED7	12040,39	1156,02	466,78
ED8	12058,82	1156,27	529,05
ED9	12077,81	1156,26	530,44
ED10	12087,73	1156,63	484,26
ED11	10176,69	1009,43	24,68
ED12	9972,54	979,83	5,68
ED13	10196,21	1011,63	12,41
ED14	10167,64	1013,71	4,49
ED15	9971,23	983,29	5,94
ED16	10152,62	1007,65	149,31
ED17	8897,33	864,35	3,73
ED18	10133,52	1007,33	115,96
ED19	9332,98	915,92	4,62
ED20	8907,15	865,1	3,88

Fonte: Autoria própria.

Em 100 itens da mochila o AG6 obteve a maior média para o genético, este algoritmo usa a mesma estratégia do AG4, porém o AG6 possui uma taxa de mutação maior, sendo de 20%. Entre as ED, ED 10 e ED13 melhores médias. De acordo com Figura 17 os ED6, ED8 e ED10 apresentam resultados semelhantes com menor grau de dispersão. Nota-se que a versão booleana (ED11 a ED20) o nível de dispersão é maior em relação a versão binária.

Figura 17 – Boxplot para mochila de 100 dimensões.



Fonte: autoria própria.

6.2.3 250 dimensões para problema da mochila

Para 250 itens a capacidade da mochila é 719.

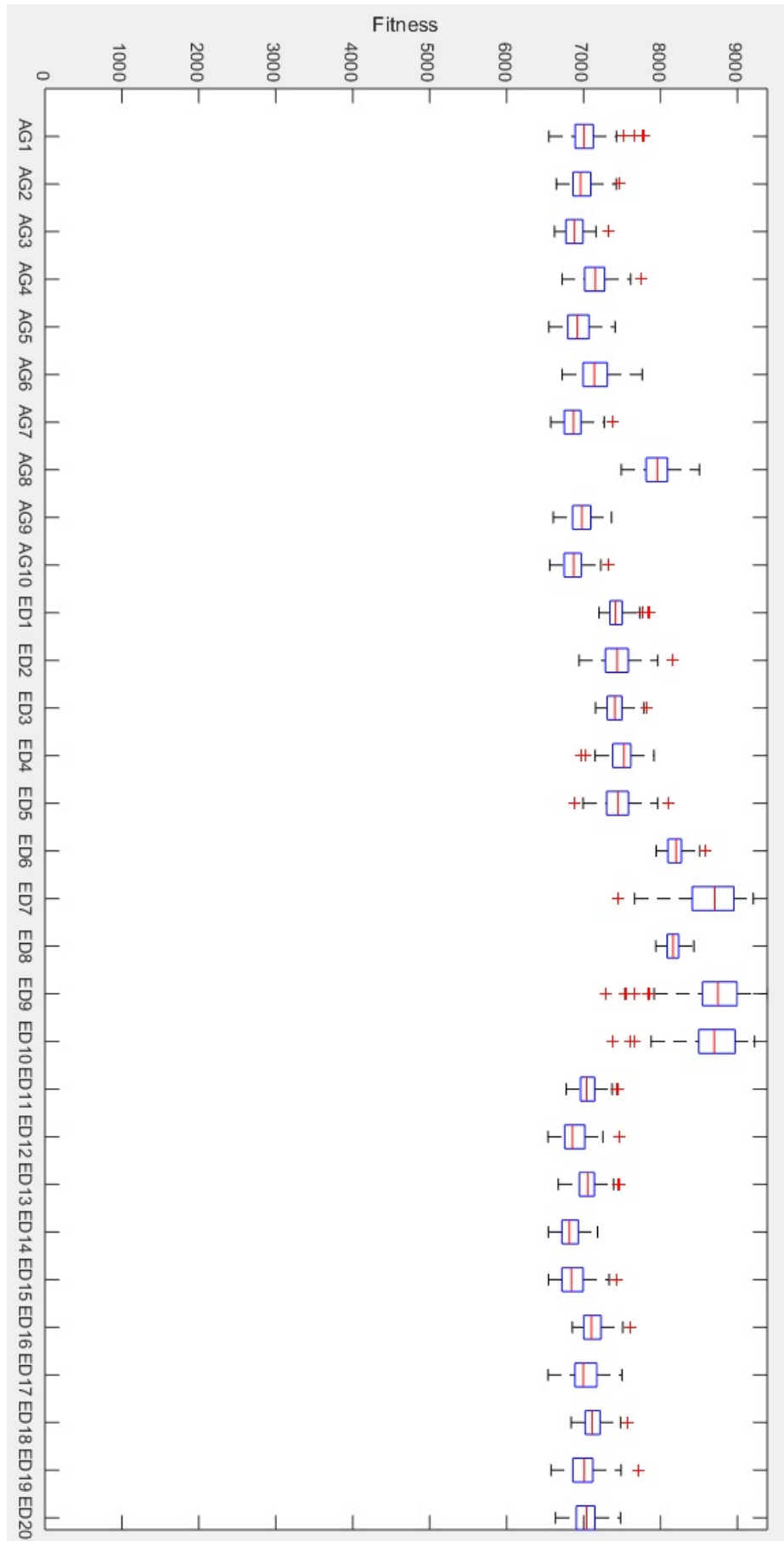
Tabela 9 – Média de convergência para 250 dimensões *Knapsack Problem*

Algoritmo	Valor	Peso	Iteração
AG1	7019,69	705,81	102,82
AG2	6982,03	705,97	59,23
AG3	6884,75	704,39	2,13
AG4	7157,56	717,64	233,84
AG5	6932,37	706,96	7,44
AG6	7162,60	717,77	233,32
AG7	6876,23	702,79	1,78
AG8	7953,17	715,32	848,34
AG9	6976,71	703,08	2,41
AG10	6871,20	702,33	1,39
ED1	7434,11	705,94	469,82
ED2	7424,5	706,29	430,51
ED3	7411,41	706,83	471,94
ED4	7499,83	705,56	388,92
ED5	7444,05	707,16	438,14
ED6	8196,79	717,08	844,26
ED7	8636,03	718,61	867,06
ED8	8164,83	716,70	864,29
ED9	8702,30	718,36	885,62
ED10	8673,49	718,45	837,47
ED11	7058,81	713,14	73,52
ED12	6882,53	704,73	1,07
ED13	7043,65	712,17	61,89
ED14	6829,17	702,48	1,00
ED15	6876,22	704,08	1,14
ED16	7120,30	718,81	103,27
ED17	7018,75	717,05	2,21
ED18	7126,13	718,40	135,22
ED19	7003,16	716,00	2,13
ED20	7051,70	716,83	2,32

Fonte: Autoria própria.

Em relação a 250 itens, o AG8 obteve melhor resultado, lembrando que esta usa da variação da roleta, enquanto que em dimensões menores os destaques para os AGs utilizaram o torneio com reposição na seleção. A ED9 e ED18 alcançaram maior média no valor de *fitness* entre suas versões. Como mostra a Figura 18, os ED6 ao ED10 tem maiores níveis de dispersão, isso significa que exploraram mais a superfície da função, logo alcançaram resultados melhores em relação aos outros algoritmos.

Figura 18 – Boxplot para mochila de 250 dimensões.



Fonte: autoria própria.

6.2.4 500 dimensões para problema da mochila

Para 500 itens a capacidade da mochila é 3422.

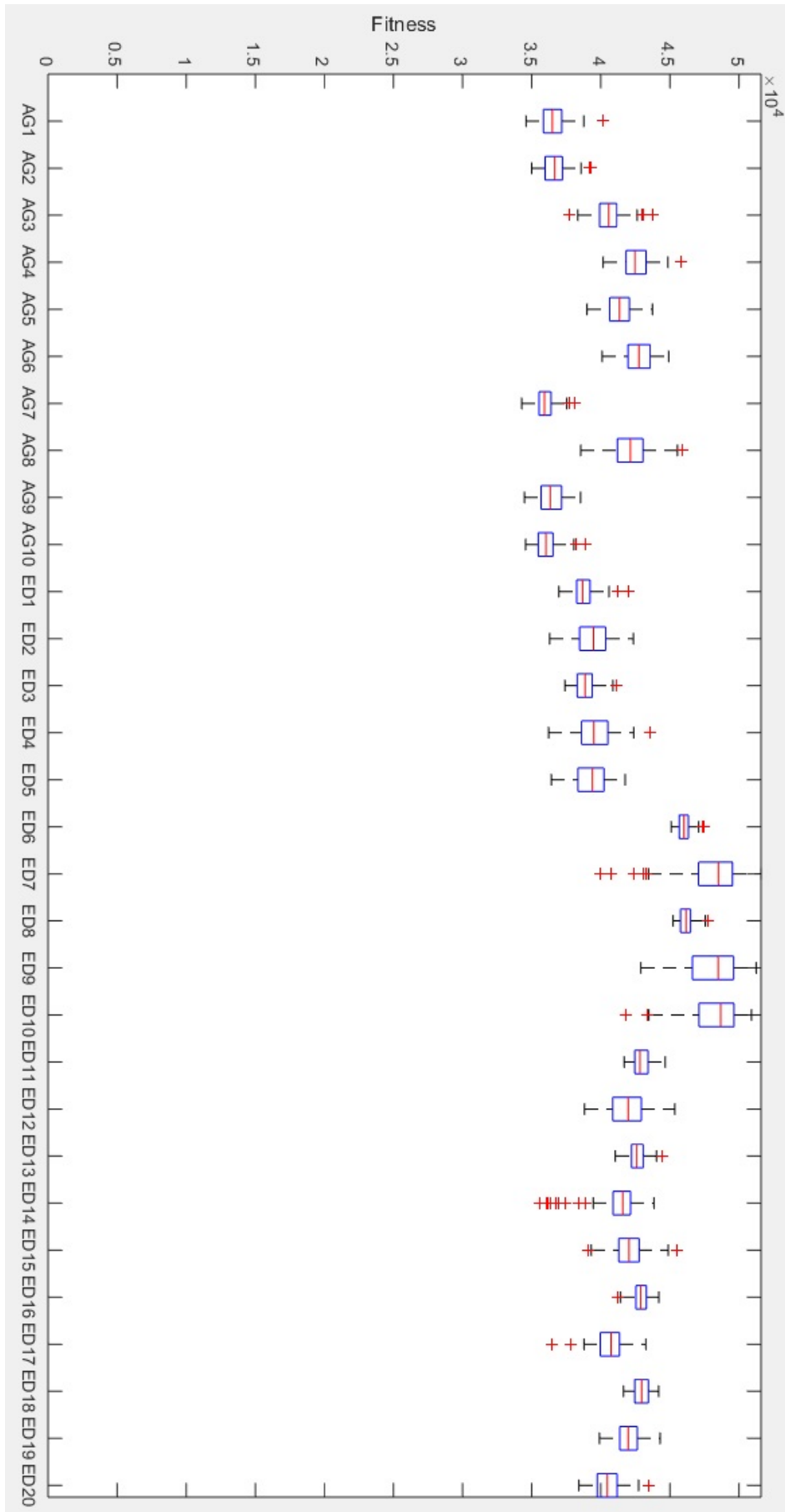
Tabela 10 – Média de convergência para 500 dimensões da mochila

Algoritmo	Valor	Peso	Iteração
AG1	36577,09	2963,81	17,33
AG2	36670,31	2981,71	52,78
AG3	40585,87	3416,12	407,93
AG4	42567,88	3421,88	591,97
AG5	41422,26	3416,37	542,61
AG6	42747,40	3421,65	598,86
AG7	35954,79	2933,54	1,86
AG8	42222,54	3138,97	876,24
AG9	36367,42	2928,64	2,19
AG10	36067,98	2942,58	1,57
ED1	38817,26	3058,76	400,7
ED2	39429,02	3099,22	453,14
ED3	38869,90	3066,58	454,54
ED4	39563,64	3131,02	442,33
ED5	39325,39	3086,20	351,48
ED6	46039,96	3414,91	862,52
ED7	48066,79	3419,94	863,42
ED8	46179,01	3414,64	875,73
ED9	48126,97	3420,21	875,82
ED10	48263,54	3419,68	837,47
ED11	42940,88	3417,11	134,80
ED12	41929,78	3399,76	2,95
ED13	42635,85	3406,60	64,97
ED14	41175,10	3356,27	1,92
ED15	42090,12	3405,41	3,45
ED16	42920,47	3421,37	120,23
ED17	40671,86	3308,79	5,02
ED18	42958,63	3421,27	131,28
ED19	41965,30	3414,25	4,53
ED20	40529,60	3335,62	4,98

Fonte: Autorial própria.

O AG6, ED10 e ED18, possuem as maiores médias de valores entre suas versões, nota-se que as médias de iterações são altas para a Evolução Diferencial binária, ED6 ao ED10. Essas variações utilizaram a recombinação exponencial, consequentemente obteve-se uma evolução significativa do *fitness* ao decorrer das gerações. O ED7 apresentou maior dispersão (Figura 19) em relação aos outros algoritmos. Os ED6 e ED8, apresentaram resultados semelhantes, com menor dispersão, tem-se em comum a mutação do tipo *Rand*, enquanto que os ED7 e ED9 usam a do tipo *Best*, mostrando que essa mutação foi relevante em relação as outras.

Figura 19 – Boxplot para mochila de 500 dimensões.



Fonte: autoria própria.

6.3 DISCUSSÃO DOS RESULTADOS

As EDs de índice 1 a 10 são versões binárias, que fazem uso de função de transferência para binarizar os vetores mutados, enquanto da ED11 a ED20 são as booleanas.

O AG8 obteve os melhores resultados entre todas as versões, com médias de *fitness* mais altas e menor grau de dispersão. Esta versão fez o uso da variação da *Stochastic Universal Sampling*, revelando a importância de se manter a variabilidade genética da população.

Os algoritmos que usaram da roleta, como os AG1 e AG2, não desempenharam muito bem, isso se deve a pressão seletiva que a roleta exerce na população, perdendo soluções ao longo das gerações.

Nos AG7 e AG10, que utilizaram o torneio da morte, pode-se observar uma menor taxa para os valores de *fitness* em relação aos seus equivalentes que usam torneio binário com reposição, como nas estratégias dos AG4, AG5 e AG6. Isso ocorre, pois o torneio da morte tem uma desvantagem desta variação com a redução de diversidade, pois cada indivíduo é uma solução, logo eliminado um indivíduo, exclui-se uma possível solução.

A mutação variável aplicada nos AG9 e AG10 levou a um desempenho superior que os AG1 e AG2, em relação a todos que estavam com a taxa de mutação fixa. A mutação é um operador que busca aumentar a diversidade genética, com uma taxa de mutação sendo crescente, aumenta-se as chances dessa variabilidade genética crescer, tendo como resultado um melhor desempenho.

Para todas as dimensões do *OneMax*, os ED binários, se destacaram por seu desempenho superior. Estes utilizaram a recombinação binomial, enquanto os demais a recombinação exponencial (ED6 a ED10). A recombinação binomial teve melhor desempenho por alterar mais componentes do vetor, esse é um fator positivo visando a maximização, pois consegue obter um vetor com a maior quantidade de bits 1.

Os genéticos AG3 ao AG6 e AG8, alcançaram performances parecidos com ED binário (ED1 ao ED10) e melhores que ED booleano (ED11 ao ED20), se tratando do problema da mochila.

Este resultado é particularmente interessante para o Algoritmo Genético, pois mostra a dificuldade imposta pelo problema. Enquanto o AG8 foi superior em todos os

casos do problema anterior, neste ele foi melhor apenas para $D = 250$. Mais relevante ainda é observar que as 3 propostas de AGs possuem características distintas em relação ao método de seleção e mutação.

As propostas de Evolução Diferencial que aplicam duas diferenças ponderadas em suas mutações, foram as que obtiveram maiores médias para as dimensões da mochila de 100, 250 e 500 itens. Nesse tipo de mutação são utilizados cinco indivíduos, dois indivíduos para cada diferença ponderada e o *Target Vector*, logo tem-se mais genes para se fazer a recombinação entre eles, possibilitando a ter um melhor desempenho em relação as que utilizam apenas uma diferença ponderada.

Já com 100, 250 e 500 itens, os algoritmos ED6 ao ED10 desempenharam melhor que todos os demais, cada um para um tamanho de problema. O que se faz relevante é destacar que todos eles utilizam de função de transferência. De certa forma, para a Evolução Diferencial Binária que usou a recombinação exponencial, não sofreu muitas alterações em todo os componentes do indivíduo e sim em intervalos específicos, funcionando como um "ajuste fino" para melhorar a combinação e obtendo melhor resultado no problema da mochila.

Abordando os dois problemas, fica nítido que a Evolução Diferencial Binária obteve melhor desempenho geral em relação as versões booleanas e ao Algoritmo Genético. Diferenciando em sua estratégia de recombinação, para o *OneMax Problem*, o binomial (ED1 ao ED5) se destacou, enquanto que na mochila o exponencial (ED6 ao ED10) alcançou resultados superiores.

Além disso, fica claro que algumas versões ED booleanas e do AG convergiram prematuramente, o que mostra que foram incapazes de fugir de mínimos locais.

7 CONCLUSÃO

Este trabalho apresentou um estudo sobre a aplicação de Algoritmos Evolutivos, a saber, Algoritmo Genético (AG) e Evolução Diferencial (DE), para otimização de problemas binários. Foram abordados o *OneMax Problem* e o problema da mochila binário.

A ideia de utilizar tais algoritmos ocorre por se tratarem de métodos de relativa facilidade de implementação e alcançarem bons resultados para os problemas correlatos.

No caso do AG, 10 versões foram elaboradas, com variações em relação à seleção dos indivíduos, método de recombinação e mutação e a presença de uma subpopulação. Para o ED, 20 alternativas foram implementadas, as quais variaram em versões binárias, que se utilizam de funções de transferência, e booleanas, nas quais as operações ocorrem pelo uso de portas lógicas. Ademais, tais propostas alteram a forma como são desenvolvidas a mutação e a recombinação.

Os resultados computacionais para o *OneMax Problem* com 50, 100, 250 e 500 dimensões relevaram que as versões da Evolução Diferencial Binária que utilizou de recombinação binomial, atingiram o máximo de resultados esperado com o valor de *fitness* igual a 1.

De forma semelhante, para o problema da mochila binário com 30, 100, 250 e 500 itens, observa-se que a Evolução Diferencial Binária obteve melhores resultados utilizando da estratégia exponencial em sua recombinação. Em síntese, os melhores resultados encontrados foram favoráveis a Evolução Diferencial Binária.

Em futuros trabalhos, pode-se utilizar outras função de transferência para a ED como a *V-shaped*, além de um estudo comparativo com outras metodologias bio-inspiradas, como algoritmos de Inteligência de Enxame. Outra possibilidade é utilizar no AG a mutação variável na roleta e utilizar outras estratégias de recombinação como a de 2 pontos. Ademais, problemas reais de seleção de variáveis merecem ser discutidos à luz das ideias aqui apresentadas.

REFERÊNCIAS

- ANDRÉ, Leanderson; PARPINELLI, Rafael Stubs. The multiple knapsack problem approached by a binary differential evolution algorithm with adaptive parameters. **Polibits**, Instituto Politécnico Nacional, Centro de Innovación y Desarrollo . . . , n. 51, p. 47–54, 2015. Citado na página 33.
- BÄCK, Thomas; FOGEL, David B; MICHALEWICZ, Zbigniew. **Evolutionary computation 1: Basic algorithms and operators**. [S.l.]: CRC press, 2018. Citado 3 vezes nas páginas 23, 24 e 26.
- CAMPOS, Alexandre de; SANTOS, Andréa MG dos; XAVIER, Gilberto F. A consciência como fruto da evolução e do funcionamento do sistema nervoso. **Psicologia USP**, SciELO Brasil, v. 8, n. 2, p. 181–226, 1997. Citado na página 13.
- CASTRO, Leandro Nunes De. **Fundamentals of natural computing: basic concepts, algorithms, and applications**. [S.l.]: Chapman and Hall/CRC, 2006. Citado 5 vezes nas páginas 13, 18, 19, 22 e 24.
- CHATTOPADHYAY, Souti; MISHRA, Sourav; GOSWAMI, Saptarsi. Feature selection using differential evolution with binary mutation scheme. In: IEEE. **2016 International Conference on Microelectronics, Computing and Communications (MicroCom)**. [S.l.], 2016. p. 1–6. Citado na página 28.
- COTTA, Carlos; TROYA, José M. A hybrid genetic algorithm for the 0–1 multiple knapsack problem. In: SPRINGER. **Artificial neural nets and genetic algorithms**. [S.l.], 1998. p. 250–254. Citado na página 34.
- COUTO, Deylon CF; SILVA, Carlos A; BARSANTE, Lillia S. Otimização de funções multimodais via técnica de inteligência computacional baseada em colônia de vaga-lumes. In: **Proceedings of the XXXVI Iberian Latin American Congress on Computational Methods in Engineering**. [S.l.: s.n.], 2015. Citado na página 33.
- DARWIN, Charles; MIRANDA, Daniel Moreira; BIZZO, Nelio. **A origem das espécies**. [S.l.]: EDIPRO, 2019. Citado 2 vezes nas páginas 16 e 17.

DENG, Changshou et al. Binary differential evolution algorithm with new mutation operator. In: IEEE. **2010 IEEE International Conference on Intelligent Computing and Intelligent Systems**. [S.l.], 2010. v. 1, p. 498–501. Citado na página 29.

DIETERICH, Johannes M; HARTKE, Bernd. Empirical review of standard benchmark functions using evolutionary global optimization. **arXiv preprint arXiv:1207.4318**, 2012. Citado na página 33.

DU, Yifei et al. Runtime analysis of onemax problem in genetic algorithm. **Journal of Robotics, Networking and Artificial Life**, Atlantis Press, v. 1, n. 3, p. 225–230, 2014. Citado na página 33.

FARIAS, Leonel Marques; SELKITTO, Miguel Afonso. Uso da energia ao longo da história: evolução e perspectivas futuras. **Revista Liberato**, v. 12, n. 17, p. 07–16, 2011. Citado na página 13.

FILHO, Marco Antonio Itaborahy. Trabalho de Conclusão de Curso, **Análise de algoritmos genéticos e evolução diferencial para otimização de funções não-lineares multimodais**. 2018. Citado na página 24.

GRIFFITHS, Anthony JF et al. Introdução à genética. In: **Introdução à genética**. [S.l.: s.n.], 2006. Citado 3 vezes nas páginas 19, 20 e 21.

HOLLAND, John Henry et al. **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence**. [S.l.]: MIT press, 1992. Citado 5 vezes nas páginas 14, 18, 20, 21 e 25.

KANNAN, S; SLOCHANAL, S Mary Raja; PADHY, Narayana Prasad. Application and comparison of metaheuristic techniques to generation expansion planning problem. **IEEE transactions on Power Systems**, IEEE, v. 20, n. 1, p. 466–475, 2005. Citado na página 13.

KONZEN, Pedro Henrique de Almeida et al. Otimização de métodos de controle de qualidade de fármacos usando algoritmo genético e busca tabu. **Pesquisa operacional**, SciELO Brasil, v. 23, n. 1, p. 189–207, 2003. Citado na página 21.

KUCHARAVY, Dmitry; GUIO, Roland De. Application of s-shaped curves. **Procedia Engineering**, Elsevier, v. 9, p. 559–572, 2011. Citado na página 29.

LINDEN, Ricardo. **Algoritmos genéticos (2a edição)**. [S.l.]: Brasport, 2008. Citado 4 vezes nas páginas 13, 14, 20 e 22.

LOPES, Sonia; ROSSO, Sergio. **Biologia. 1ª edição**. [S.l.]: Vol. único. São Paulo: Editora Saraiva, 2005. Citado 2 vezes nas páginas 18 e 20.

MAGALHAES-MENDES, Jorge. A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem. **WSEAS transactions on computers**, v. 12, n. 4, p. 164–173, 2013. Citado na página 33.

MICHALEWICZ, Zbigniew; HARTLEY, Stephen J. Genetic algorithms+ data structures= evolution programs. **Mathematical Intelligencer**, Berlin; New York: Springer-Verlag,[c1978-, v. 18, n. 3, p. 71, 1996. Citado 2 vezes nas páginas 22 e 23.

OPARA, Karol R; ARABAS, Jarosław. Differential evolution: A survey of theoretical analyses. **Swarm and evolutionary computation**, Elsevier, v. 44, p. 546–558, 2019. Citado na página 30.

PRICE, Kenneth; STORN, Rainer M; LAMPINEN, Jouni A. **Differential evolution: a practical approach to global optimization**. [S.l.]: Springer Science & Business Media, 2006. Citado 3 vezes nas páginas 20, 28 e 29.

SILVA, Fábio Portela da et al. Estudo comparativo do desempenho de diferentes operadores genéticos na resolução do problema do caixeiro viajante. In: **Colloquium Exactarum**. [S.l.: s.n.], 2017. v. 9, n. 1. Citado na página 33.

SINGH, Mahavir; TYAGI, Suresh C. Genes and genetics in eye diseases: a genomic medicine approach for investigating hereditary and inflammatory ocular disorders. **International journal of ophthalmology**, Press of International Journal of Ophthalmology, v. 11, n. 1, p. 117, 2018. Citado na página 20.

SINGH, Rattan Preet. Solving 0–1 knapsack problem using genetic algorithms. In: IEEE. **2011 IEEE 3rd International Conference on Communication Software and Networks**. [S.l.], 2011. p. 591–595. Citado na página 34.

VON, Zuben; FERNANDO, J. Computação evolutiva: uma abordagem pragmática. **Anais da I Jornada de Estudos em Computação de Piracicaba e Região (1a JECOMP)**, v. 1, p. 25–45, 2000. Citado na página 16.

YANASSE, Horacio Hideki; SOMA, Nei Yoshihiro. A new enumeration scheme for the knapsack problem. **Discrete applied mathematics**, Elsevier, v. 18, n. 2, p. 235–245, 1987. Citado na página 35.

ZHANG, L et al. Design of planar thinned arrays using a boolean differential evolution algorithm. **IET microwaves, antennas & propagation**, IET, v. 4, n. 12, p. 2172–2178, 2010. Citado na página 28.