

Artigo de Revisão

EVOLUÇÃO DIFERENCIAL BINÁRIA: UMA REVISÃO DE LITERATURA

Binary differential evolution: a literature review

**Jade Diane Fernandes Targino Filgueira^{1*}, Hugo Valadares Siqueira², Carmelo J. A. Bastos-Filho³,
Adriano Diniz Pereira⁴**

¹Mestre, Programa de Pós-Graduação em Engenharia de Produção da Universidade Tecnológica Federal do Paraná, Ponta Grossa, Brasil. Prof. do Instituto Federal de Educação, Ciência e Tecnologia do Tocantins, *campus* Araguaína (IFTO).

²Doutor, Programa de Pós-Graduação em Engenharia de Produção da Universidade Tecnológica Federal do Paraná, Ponta Grossa, Brasil.

³Doutor, Programa de Pós-Graduação em Engenharia de Computação, Universidade de Pernambuco, Recife, Brasil.

⁴Especialista, Programa de Pós-Graduação em Engenharia de Produção da Universidade Tecnológica Federal do Paraná, Ponta Grossa, Brasil.

Submetido em: 18.12.2024; Aceito em: 30.05.2025; Publicado em: 10.10.2025

***Autor para correspondência:** jade.targino@ifrj.edu.br

Resumo: Com o objetivo de entender o que tem sido pesquisado no campo de meta-heurísticas, particularmente no que diz respeito ao método de Evolução Diferencial aplicado a problemas de natureza binária, foi desenvolvida uma revisão de literatura, utilizando o termo de busca “Binary Differential Evolution”. Os 32 artigos analisados foram categorizados em cinco tópicos: uso de operadores lógicos, utilização de funções de transferência, utilização de operadores de estimativa de probabilidade, meta-heurísticas combinadas e operadores de mutação não-paramétricos. As análises feitas revelaram uma mudança dos objetivos das publicações anteriormente voltada para a criação de mecanismos inovadores de binarização passando a apresentar, nos anos mais recentes, maior preocupação em melhorar a performance do método, muitas vezes recorrendo à hibridização com outros métodos. Outra tendência observada a de desenvolver algoritmos específicos para determinadas aplicações. A análise e categorização do conteúdo dos trabalhos encontrados pode servir como um guia para pesquisadores que desejem utilizar o método de ED aplicado a problemas de natureza binária ou estudar novas maneiras de fazê-lo.

Palavras-chave: Problemas de Otimização Binária, Evolução Diferencial Binária, Meta-heurística.

Abstract: In order to understand what has been researched in the field of metaheuristics, particularly with regard to the Differential Evolution method applied to binary problems, a literature review was developed using the search term “Binary Differential Evolution”. The 32 articles analyzed were categorized into five topics: use of logical operators, use of transfer functions, use of probability estimation operators, combined metaheuristics and non-parametric mutation operators. The analyses performed revealed a change in the objectives of the publications, which were previously focused on the creation of innovative binarization mechanisms, and in more recent years have become more concerned with improving the performance of the method, often resorting to hybridization with other methods. Another trend observed was the development of specific algorithms for certain applications. The analysis and categorization of the content of the works found can serve as a guide for researchers who wish to use the DE method applied to binary problems or study new ways of doing so.

Keywords: Binary Optimization Problems, Binary Differential Evolution, Meta-heuristics.

INTRODUÇÃO

A Evolução Diferencial (ED) é um método estocástico e populacional de otimização global desenvolvido para resolver problemas de otimização contínua. Desde 1995, a ED e suas variantes aparecem como uma das classes de algoritmos Evolucionários (EAs) mais competitivos e adaptáveis e tem sido efetivamente implementada para resolver problemas do mundo real (DHALIWAL & DHILLON, 2022).

A ED é simples na execução e configuração de parâmetros, sendo necessários apenas alguns parâmetros de controle obrigatórios para atualizar a população (DHALIWAL & DHILLON, 2019). A ED é bem estabelecida na literatura de pesquisa operacional e otimização, sendo amplamente utilizada para uma vasta gama de problemas de otimização contínua devido à sua simplicidade, eficácia e robustez.

A Evolução Diferencial Binária (EDB) é uma variante do algoritmo de ED, que adapta os princípios da ED para lidar especificamente com espaços de soluções discretas, particularmente problemas de otimização binária. Essa adaptação é crucial em diversos campos de aplicação, como otimização de redes, programação de horários, e problemas de decisão financeira, onde as variáveis de decisão são frequentemente restritas a valores binários, indicando a seleção ou rejeição de opções, ativação ou desativação de elementos, entre outras.

O objetivo do presente estudo é fazer um levantamento a respeito das diferentes técnicas utilizadas para adaptar o método da ED a problemas de natureza binária, fornecendo um apanhado para pesquisadores que tenham interesse em aplicar a EDB ou em desenvolver novas variantes binárias da ED o que pode ser necessário à medida em que se descubrem novas aplicações para a EDB.

Este artigo está dividido em 04 seções. A primeira é a presente seção, introdução, onde é contextualizado o tema geral do trabalho e descrito o objetivo de sua realização. A segunda seção, Metodologia, descreve os procedimentos utilizados para sua realização. A seção de Desenvolvimento contém uma breve descrição da ED e as análises dos artigos selecionados e a seção de Conclusão encerra a pesquisa e sumariza suas contribuições.

MATERIAL E MÉTODOS

Foi feita uma revisão de literatura, buscando conhecer diferentes adaptações do algoritmo original da Evolução Diferencial para tornar sua utilização adequada a problemas binários. A busca foi feita através da consulta a 4 bases de dados internacionais de prestígio: ACM Association for Computing Machinery, IEEE Xplore, Science Direct e Emerald Insight.

Cada uma das bases foi consultada com as seguintes palavras-chave: “Binary Differential Evolution”. A busca foi feita nos resumos nas bases ACM, IEEE Xplore, e Emerald Insight. Como não foi encontrada essa opção na *interface* da Science Direct, optou-se pela opção “Título, Resumo e Palavras-chave”, retornando na busca feita em 18/08/2023, um total de 94 resultados, conforme detalhado na Tabela 1:

Tabela 1. Resultados da Pesquisa Bibliográfica Inicial. Fonte: Autoria Própria.

	Science Direct	IEE Xplore	ACM	Emerald Insight	Total
Busca	Título, Resumo e Palavras-Chave	Resumo	Resumo	Resumo	
"Binary Differential Evolution"	44	44	5	1	94

Após a primeira busca, identificou-se que as bases Science Direct e IEEE Xplore eram as mais significativas em número de publicações que se encaixavam dentro dos critérios de busca, por isso optou-se por prosseguir utilizando apenas os resultados das duas bases de dados. Assim, obtendo um total de 8 publicações, conforme detalhado na Tabela 2.

Tabela 2. Resultado pela busca BABC e BDE. Fonte: Autoria Própria.

Base	Science Direct	IEE Xplore	
Busca	Título, Resumo e Palavras-Chave	Resumo	Total
"Binary Differential Evolution"	44	44	88

Os trabalhos encontrados foram analisados e foram incluídos na pesquisa todos aqueles que demonstrassem alguma forma inovadora de transformar os métodos contínuos em binários, totalizando 32 artigos. Foram descartados os trabalhos que não atendessem o critério de inclusão, totalizando 56 artigos. Os artigos que foram desconsiderados para os propósitos do presente trabalho se enquadravam em alguma das seguintes situações: Apesar de conter os termos de busca em algum dos campos selecionados, utiliza o método de ED apenas como parâmetro de comparação com outros métodos; utiliza o método em alguma aplicação sem modificar nenhum aspecto do mesmo.

DESENVOLVIMENTO

A Evolução Diferencial (ED) é uma técnica de computação evolucionária que foi originalmente desenvolvida por Storn e Price (1997). O algoritmo da ED é inicializado gerando uma população aleatória de indivíduos $x_{i,G} = x_{ij,G}$, onde $i = 1, 2, \dots, NP$, sendo NP o número de indivíduos na população, G é o índice que corresponde a geração atual, $j = 1, 2, \dots, D$, sendo D o número de dimensões do problema. Após a inicialização, o algoritmo segue procurando pela solução ótima, atualizando as gerações dos indivíduos através das operações de Mutação, Recombinação e Seleção.

A etapa de mutação é responsável pela perturbação da população atual através da adição da diferença ponderada entre os genes de dois cromossomos selecionados aleatoriamente aos genes de um outro cromossomo da população. Dessa forma, para cada indivíduo $x_{i,G}$ um vetor mutante $v_{i,G+1}$ é gerado de acordo com a equação 1:

$$v_{i,G+1} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G}) \quad (1)$$

Onde os índices $r1, r2$ e $r3 \in \{1, 2, \dots, NP\}$ são escolhidos aleatoriamente e devem ser diferentes entre si e também de i , dessa forma o número de vetores parâmetro em uma população deve ser no mínimo quatro. F é um número real pertencente ao intervalo entre 0 e 2, sem incluir o 0 que controla a amplificação do vetor diferença $(x_{r2,G} - x_{r3,G})$.

Para manter a diversidade dentro da população perturbada, e mesclar informações antigas e novas, após a mutação, o novo vetor é modificado através do processo de recombinação, onde o vetor alvo é combinado com o vetor mutado usando o seguinte esquema demonstrado na equação 2 para resultar no vetor de teste $u_{i,G+1} = (u_{1i,G+1}, u_{2i,G+1}, \dots, u_{Di,G+1})$:

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{se } (r(j) \leq CR) \text{ ou } j = rn(i) \\ x_{ji,G} & \text{se } (r(j) > CR) \text{ ou } j \neq rn(i) \end{cases} \quad (2)$$

Para $j=1,2,\dots,D$. $r(j) \in [0,1]$ é a j -ésima avaliação de um gerador uniforme de números aleatórios. CR é a taxa de recombinação (*Crossover Rate*) ($CR \in [0,1]$). $rn(i) \in (1,2,\dots,D)$ é um índice escolhido aleatoriamente que assegura que $u_{i,G+1}$ recebe pelo menos um elemento de $v_{i,G+1}$ para garantir que novos vetores sejam produzidos e a população se altere.

Na etapa de seleção, um esquema de seleção ambiciosa é usado: Se o vetor de teste resulta em uma função custo melhor comparado ao vetor parâmetro $x_{i,G}$, ele é aceitado como novo vetor pai para a geração seguinte $G+1$. Caso contrário, o vetor alvo é mantido para servir como pai para a geração $G+1$ novamente.

O algoritmo ED original não pode ser diretamente implementado para resolver problemas de natureza binária porque foi originalmente desenvolvido para resolver problemas de otimização de domínio contínuo. Para lidar com variáveis de decisão binárias, há necessidade de algumas mudanças. A inicialização e a mutação são as duas etapas que diferenciam o BDE da DE, pois a saída das etapas de inicialização e mutação deve sempre estar de acordo com o tipo de variável do problema (DHALIWAL & DHILLON, 2022).

Nesse sentido, muitos autores têm contribuído acerca de métodos capazes de adaptar a ED para problemas envolvendo variáveis de natureza binária em diversos campos de aplicação. A seguir uma revisão de literatura é apresentada sobre a implementação de diferentes métodos binários para a ED. Os artigos analisados foram categorizados conforme os mecanismos utilizados para a adaptação do método a espaços de busca binários. As categorias encontradas durante a análise dos artigos selecionados foram: uso de operadores lógicos, utilização de funções de transferência, utilização de operadores de estimativa de probabilidade, meta-heurísticas combinadas e operadores de mutação não-paramétricos. A Tabela 3 apresenta os 32 artigos por meio de sua referência, categoria e também sua respectiva área de aplicação. É importante esclarecer que muitos autores utilizam instâncias de problemas padrão conhecidos na literatura para validar seus métodos, esse tipo de aplicação é conhecido como um problema *benchmark* e existem vários tipos conhecidos.

Operadores lógicos

He e Han (2007) propuseram um novo algoritmo de Evolução Diferencial Binária baseado na teoria da imunidade da biologia, para os autores, apesar da simplicidade da técnica, simulações mostram melhoria na habilidade de incrementação e velocidade de convergência em comparação com outros métodos.

O algoritmo proposto consiste inicialmente em uma modificação na etapa de mutação da ED básica. As operações de soma, multiplicação e subtração da equação original de mutação são substituídas pelos operadores de porta lógica “or”, “and” e “xor”, respectivamente.

Em seguida, os vetores mutantes gerados passarão por um processo de seleção baseado na teoria da imunidade, onde um valor limite é estabelecido V_{max} esse valor deve estar dentro do intervalo $(1 \leq V_{max} \leq D)$.

A cada vetor mutante deve ser atribuído um valor $L(v|i, G+1) = \sum_{j=1}^D v_{ji, G+1}$. Caso $L(v|i, G+1)$ seja menor ou igual a V_{max} assume-se que o vetor mutante não é alto antígeno e por isso pode ser utilizado nos passos seguintes. Caso contrário, o vetor é alto-antígeno e deve sofrer uma manipulação para reduzir o seu comprimento.

Tabela 3. Categorização dos resultados e respectivas áreas de aplicação. Fonte: Autoria Própria.

	Referência	Categoria	Área de Aplicação
1	He e Han (2007)	Operadores Lógicos	Problemas <i>Benchmark</i>
2	Deng <i>et al.</i> (2008)	Funções de transferência	Problemas <i>Benchmark</i>
3	Deng <i>et al.</i> (2009)	Funções de transferência	Problemas <i>Benchmark</i>
4	He <i>et al.</i> (2009)	Funções de transferência	Seleção de subconjunto de características
5	Deng <i>et al.</i> (2010)	Funções de transferência	Problemas <i>Benchmark</i>
6	Wu e Tseng (2010)	Operadores Lógicos	Otimização topológica de estruturas
7	Singh <i>et al.</i> (2012)	Funções de transferência	Problema de alocação de canais
8	Wang e Guo (2012)	Operadores Lógicos	Aprendizado de Redes Baeysianas
9	Wang <i>et al.</i> (2012)	Operadores de estimativa de probabilidade	Problemas <i>Benchmark</i>
10	Krause e Lopes (2013)	Funções de transferência	Problemas <i>Benchmark</i>
11	Parija <i>et al.</i> (2014)	Operadores de estimativa de probabilidade	Redução de Custos em redes móveis
12	Chen <i>et al.</i> (2015)	Meta-heurísticas combinadas	Problema de comprometimento de Unidade Térmica
13	Li e Zhang (2015)	Operadores Lógicos	Problema de Otimização de dois níveis
14	Banitalebi <i>et al.</i> (2016)	Operadores Lógicos	Problemas <i>Benchmark</i>
15	Fan e Huang (2017)	Operadores Lógicos	Alocação de recursos
16	Zhang e Li (2017)	Meta-heurísticas combinadas	Problemas <i>Benchmark</i>
17	Dhalwal e Dhillon (2018)	Operadores de estimativa de probabilidade	Problema de comprometimento de Unidade Térmica baseado no lucro
18	Bidgoli <i>et al.</i> (2019)	Operadores de mutação não-paramétricos	Seleção de características
19	He <i>et al.</i> (2019)	Operadores de mutação não-paramétricos	Minimização lógica
20	Li <i>et al.</i> (2019)	Operadores de mutação não-paramétricos	Otimização de subconjunto de características
21	Li e Dong (2019)	Operadores de mutação não-paramétricos	Otimização de subconjunto de características e agrupamento
22	Zhang <i>et al.</i> (2020)	Operadores de mutação não-paramétricos	Seleção de características com objetivos múltiplos
23	Ali <i>et al.</i> (2021)	Funções de transferência	Problemas <i>Benchmark</i>
24	Ghosh <i>et al.</i> (2021)	Funções de transferência	Seleção de características
25	Hsieh (2021)	Funções de transferência	Otimização de processos de produção em espaços cyber físicos
26	Tarkhaneh <i>et al.</i> (2021)	Funções de transferência	Seleção de características
27	Xie <i>et al.</i> (2021)	Funções de transferência	Seleção de características
28	Xuan <i>et al.</i> (2021)	Operadores Lógicos	Seleção de características
29	Zhang <i>et al.</i> (2021)	Operadores de mutação não-paramétricos	Seleção de características
30	Gao <i>et al.</i> (2023)	Operadores de estimativa de probabilidade	Deteção de mudanças
31	Wang <i>et al.</i> (2023)	Operadores Lógicos	Seleção de características
32	Wei <i>et al.</i> (2023)	Operadores Lógicos	Seleção de características

Wu e Tseng (2010) propuseram um novo algoritmo binário de otimização baseado na ED. Os autores criaram um operador binário de mutação construído com a utilização de operadores lógicos. O operador lógico XOR é aplicado aos vetores x_i e x_{r_1} para calcular dois vetores, um contendo os termos comuns entre x_i e x_{r_1} e o outro contendo apenas os termos diferentes. Esses dois vetores sofrerão mutação baseado em dois índices F1 e F2 e depois combinados, resultando em um vetor único mutado que passará para a etapa de recombinação.

Wu e Du (2012) utilizaram a Teoria da Ressonância Adaptativa (ART) integrada ao algoritmo da Evolução Diferencial Binária Modificado apresentado por Wu e Tseng (2010). O objetivo principal do ART no algoritmo é classificar a população em grupos com padrões semelhantes.

Wang e Guo (2012) desenvolveram um novo algoritmo utilizando a ED para tratar de Aprendizado de Redes Bayesianas. Para isso os vetores iniciais são representados por matrizes contendo apenas elementos binários, dada a natureza dos Grafos Acíclicos Direcionados Bayesianos. Além disso, operadores lógicos são utilizados no lugar das operações matemáticas na equação de mutação do algoritmo tradicional da ED e F (Fator de escala da equação de mutação) também é representado por uma matriz binária.

Li e Zhang (2015) criaram um algoritmo binário baseado na ED para tratar de problemas de otimização de dois níveis. O algoritmo utiliza codificação binária para inicializar os indivíduos que em seguida passarão pelas etapas de mutação, recombinação e seleção. Para a etapa de mutação foram criados dois operadores inéditos baseados na utilização do operador lógico XOR. Os operadores de recombinação e seleção também foram adaptados, mas seguem a mesma lógica dos consecutivos operadores na ED original.

Banitalebi *et al.* (2016) apresentaram o algoritmo denominado Evolução Diferencial Binária Alto Adaptativa (SabDE) no qual usam um mecanismo adaptativo para selecionar como novas soluções experimentais são geradas e um processo caótico para adaptar valores dos parâmetros. Para garantir que qualquer uma das estratégias de mutação da ED possa ser utilizada em espaços binários uma abordagem baseada no uso de portas lógicas é proposta.

Fan e Huang (2017) criaram um novo algoritmo que apresenta uma nova estratégia de mutação para a ED que visa torná-lo adequado a espaços binários e mais eficaz em problemas de otimização. O novo operador de mutação se baseia no uso de operadores lógicos XOR e opera em duas etapas. A primeira utiliza um parâmetro de probabilidade para controlar a aplicação de uma operação XOR entre dois vetores, o resultado desta passa por uma nova operação XOR com um outro vetor escolhido aleatoriamente.

Xuan *et al.* (2021) desenvolveram um operador de mutação modificado binário para ser introduzido no seu algoritmo discreto baseado na ED para tratar de problemas de seleção de características. O operador proposto substitui a operação de subtração pelo operador lógico XOR na equação original de mutação da ED. O resultado é comparado com um número aleatório para dar origem ao vetor após a mutação.

Wang *et al.* (2023) desenvolveram um algoritmo que contém um operador binário de mutação com técnicas de nicho. As equações do novo operador de mutação envolvem a escolha dos *bits* do vetor mutante a partir da comparação de um valor aleatório *rand* com uma constante $C_{i,j}$ que por sua vez pode assumir o valor de um parâmetro σ (coeficiente de perturbação) ou ser calculado a partir da aplicação da função XOR aos vetores $x_{r_1,j}$ e $x_{r_2,j}$ que são os vetores dominados dentre os três vetores aleatórios escolhidos para participar do processo de mutação.

Wei *et al.* (2023) também propuseram um novo operador de mutação para tratar de problemas de seleção de características utilizando ED. O operador consiste em gerar probabilidades de mutação para os bits do vetor dominante dentre os três vetores aleatórios escolhidos para participar do processo de mutação. Essas probabilidades são calculadas através de um processo aplicação da função XOR aos vetores dominados e pela multiplicação do resultado por um fator de escala F e um número aleatório *rand*.

Funções de transferência

Deng *et al.* (2008) propuseram uma função de transferência para mapear a variável x de um domínio contínuo em um domínio discreto. O intervalo $[0,1]$ foi particionado em dois subintervalos iguais. Se $x \in [0,0.5]$ então sua imagem é 0 e caso $x \in [0.5,1]$ sua imagem é 1.

Além disso, também foi necessário desenvolver um operador para garantir que o vetor gerado através de mutação e *crossover* esteja dentro do intervalo $[0,1]$. O mecanismo consiste em substituir os indivíduos que fiquem fora do intervalo desejado pelo melhor indivíduo da geração anterior.

Deng *et al.* (2009) propuseram um algoritmo para otimização discreta baseado na ED. Em primeiro lugar, para garantir que o operador de mutação original possa ser utilizado é proposto um operador para mapear as variáveis do domínio discreto para o contínuo. Então a função sigmoide é utilizada para manter o resultado do operador de mutação no intervalo $[0,1]$. Antes do operador de *crossover*, um operador de mapeamento inverso descrito em Deng *et al.* (2008) transforma os números contínuos do intervalo entre $[0,1]$ para discretos.

Para adaptar a heurística da ED para trabalhar com problemas discretos, He *et al.* (2009) desenvolveram um algoritmo no qual adicionam um operador discreto que, resumidamente, trata-se de uma função de transferência aplicada tanto aos vetores iniciais $X_{i,G}$ de cada geração quanto aos vetores mutados e recombinados $U_{i,G+1}$ antes de passar pela etapa de seleção do algoritmo básico da ED.

Deng *et al.* (2010) propuseram um novo algoritmo com alterações nas etapas de inicialização e mutação. Na etapa de inicialização, uma operação de arredondamento é adicionada para que a população inicial gerada $X = (x|ij)_{mxn}$ contenha apenas vetores binários. Na etapa de mutação, um novo operador é proposto, no qual as dimensões de cada indivíduo de X sofrem inversão com base no sorteio de dois números aleatórios *rand1* (para cada gene de todos os indivíduos da presente população) e *rand2* (para cada geração). Caso, *rand1* seja menor que *rand2* a operação de inversão é executada.

Singh *et al.* (2012) propuseram um novo algoritmo discreto, binário, baseado na ED para resolver o problema de alocação de canais em redes de telefones celulares. O procedimento de discretização consiste em uma função de transferência com uso da função sigmoide, que é aplicada a cada elemento dos vetores mutados. Caso o resultado seja maior ou igual a um número aleatório gerando entre 0 e 1, assume-se valor 1 para o elemento do vetor mutado discreto binário, caso contrário, assume-se valor 0.

Krause e Lopes (2013) apresentaram um novo algoritmo discretizado baseado na ED onde o número contínuo F em cada dimensão i do vetor gerado na fase de inicialização é aplicado à função sigmoide para discretizar o vetor de soluções. Se o resultado for maior que zero, a i -ésima dimensão é definida como 1, caso contrário, é definida como 0. Esta nova versão da Evolução Diferencial é um algoritmo codificado continuamente que discretiza apenas o vetor de solução.

A versão binária da ED dos autores Ali *et al.* (2021) conta com um mecanismo de mapeamento que transforma uma população inicial binária em uma contínua. Após passar pelos operadores tradicionais de mutação e recombinação, as soluções são remapeadas para o espaço binário através do procedimento descrito por Ali *et al.* (2019).

Ghosh *et al.* (2021) propuseram um algoritmo binário baseado na ED que conta com um operador de mutação binário que utiliza a distância de manhattan (métrica de distância que considera apenas movimentos na horizontal e vertical) ao invés da tradicional diferença entre vetores da ED original. Após multiplicar o fator de escala pelo resultado, uma operação de arredondamento é executada.

Hsieh (2021) desenvolveu um algoritmo denominado ED discreta (DDE) que segue todos os passos da ED, acrescentando uma etapa de transformação entre a recombinação e seleção, o que significa que antes de terem os fitness avaliados para que os melhores indivíduos sejam selecionados, as soluções serão transformadas em vetores binários através de uma função de transferência baseada no uso da função sigmoide.

Tarkhaneh *et al.* (2021) criaram um algoritmo para seleção de características baseados na ED. O problema de seleção de características tem natureza binária, por isso as soluções iniciais são geradas aleatoriamente e arredondados para vetores contendo apenas 0 ou 1. O *loop* principal do algoritmo acontece em duas etapas e inclui a atualização do fator de escala e a taxa de recombinação, duas estratégias de mutação originais e um operador de recombinação modificado.

Xie *et al.* (2021) atualizaram o método de cálculo do vetor diferença e aplicaram um fator de escala aleatório que é atualizado com base no progresso do número de gerações do algoritmo. Além disso, para tornar o resultado do vetor de mutação mais próximo entre 0 e 1, usaram um operador baseado na função tangente hiperbólica. Para a recombinação também é sugerido uma taxa de recombinação adaptativa para garantir diversidade nas populações iniciais e rapidez de convergência para as populações mais maduras.

Operadores de Estimativa de Probabilidade

Wang *et al.* (2012) criaram um operador de estimativa de probabilidade capaz de criar indivíduos mutantes dentro do espaço binário. O operador consiste em uma função que utiliza o Fator de escala e os indivíduos mutados através do operador de mutação da ED original e um valor real positivo chamado de fator bandwidth.

Parija *et al.* (2014) chamaram seu novo algoritmo binário baseado na ED de BDE (Binary Differential Evolution). As contribuições do BDE são: Vetores binários de RCC mutantes gerados usando um novo operador de estimativa de probabilidade (PEO). O PEO inclui duas etapas, o cálculo de estimativa de probabilidade e uma equação de binarização com base na etapa anterior. Além disso, o BDE apresenta operadores modificados para as etapas de recombinação e seleção também com base nos vetores de estimativa de probabilidade.

Dhaliwal e Dhillon (2018) apresentaram uma versão binária para a Evolução Diferencial baseada na codificação binária das soluções iniciais e na modificação do processo de mutação, empregando um Operador de Estimativa de Probabilidade (PEO) integrado ao operador de mutação da ED para produzir vetores mutantes binários.

Gao *et al.* (2023) projetaram um novo operador de mutação e uma estratégia de adaptação de geração de operador de mutação para problemas de detecção de mudanças. O operador de mutação binário proposto se baseia em um operador de estimativa de probabilidade baseado na função sigmoide. Um valor aleatório é comparado com a probabilidade de mutação para decidir se um determinado *bit* de um cromossomo sofrerá mutação para 0 ou 1.

Meta-Heurísticas Combinadas

Chen *et al.* (2015) incorporaram o mecanismo de aprendizado do PSO à ED e criaram um novo algoritmo chamado de Evolução Diferencial Binária baseada em Aprendizagem (BLDE). Ele mantém duas coleções de soluções: a população atual $X(t)$ e um arquivo chamado $A(t)$, ambas binárias. A cada iteração, novos indivíduos são criados através da combinação entre os melhores genes dos indivíduos das duas populações.

Zhang e Li (2017) criaram um algoritmo híbrido através da junção do BPSO e uma versão binária da ED proposta por eles. No BPSO a função sigmoide é utilizada para transformar as soluções finais em binárias e no BDE é utilizada codificação binária e dois operadores de mutação baseados no uso do operador lógico XOR que garantem a utilização do método dentro do espaço binário.

Operadores de Mutação Não Paramétricos

O operador de mutação da ED geralmente dá origem a vetores não binários, devido ao uso do fator de escala F que varia no intervalo entre 0 e 2 de forma contínua. Alguns autores adaptaram a ED, propondo operadores de mutação que não dependem do fator de escala como parâmetro e que por isso podem ser aplicados a vetores com codificação binária e que garantem a criação de vetores mutantes com codificação binária.

Bidgoli *et al.* (2019) desenvolveram um novo algoritmo de evolução diferencial binário multi-objetivo para seleção de características. Os vetores são inicializados com codificação binária e o operador de mutação desenvolvido não altera a codificação, como as demais etapas da ED também não interferem na codificação binária, nenhuma outra estratégia foi necessária.

O operador de mutação proposto funciona da seguinte forma: a seleção de cada recurso na nova solução candidata será determinada com base no status desse recurso em três soluções candidatas x_{r1} , x_{r2} e x_{r3} . Existem três casos relativos ao *status* de um recurso:

- 1) Se um recurso é selecionado em apenas uma das duas soluções candidatas x_{r2} e oposto de x_{r3} , ou seja, o valor do gene é 0 em um deles e 1 em outro, então o status do recurso correspondente na nova solução candidata será igual a x_{r1} .
- 2) se um recurso for selecionado em ambas as soluções selecionadas, x_{r2} e oposto de x_{r3} , então há um acordo sobre a seleção desse recurso, então ele seria selecionado na nova solução candidata com probabilidade 0,5.
- 3) se um recurso não está selecionado em ambas as soluções selecionadas, x_{r2} e oposto de x_{r3} , então existe um acordo sobre a não seleção desse recurso então não seria selecionado na nova solução candidata com probabilidade de 0,5.

He *et al.* (2019) criaram uma nova versão de ED binária através de codificação binária, um novo operador de mutação aleatório binário e um operador melhorado de seleção que escolhe os melhores indivíduos dentro de uma população formada da união entre os vetores alvo e vetores de teste, evitando que indivíduos com fitness importantes sejam deixados de lado por perderem a competição no mecanismo tradicional de seleção que leva em consideração apenas a competição de um par por vez.

O funcionamento do novo operador de mutação é demonstrado pela equação 3:

$$v_{i,j} = \begin{cases} x_{i,j} + (-1)^{x_{i,j}} \cdot |x_{r2,j} - x_{r3,j}|, & \text{se } rand \geq 0,5 \\ x_{best,j} + (-1)^{x_{best,j}} \cdot |x_{r2,j} - x_{r3,j}|, & \text{se } rand < 0,5 \end{cases} \quad (3)$$

O valor absoluto do vetor de diferença é apenas 0 ou 1, sendo assim a variável mutante ainda será binária.

Li *et al.* (2019) utilizaram a ED para criar um algoritmo binário para resolver problemas de otimização de subconjuntos de características. O Algoritmo proposto conta com uma nova estratégia de inicialização a partir de aprendizado baseado em oposição e um operador binário de mutação que os autores afirmam ser original, porém durante a análise foi constatado ser igual ao desenvolvido em He *et al.* (2019). Ambos os trabalhos datam de 2019 e podem ter sido desenvolvidos como originais simultaneamente.

Li e Dong (2019) combinaram ED binária e contínua em um novo algoritmo para seleção e agrupamento de características não supervisionados. Primeiro, uma versão binária da ED foi utilizada para seleção de características, na qual um operador de mutação binário foi desenvolvido. O operador funciona muito semelhante ao descrito pela equação 3, contudo se limita à segunda parte da equação para qualquer intervalo.

Zhang *et al.* (2020) usaram a evolução diferencial para otimizar a seleção de características com múltiplos objetivos. Para isso, desenvolveram um operador de mutação binário que seleciona o melhor entre três vetores aleatórios como vetor base e utiliza a diferença entre os outros dois vetores para determinar uma probabilidade de mutação a ser aplicada ao vetor base na geração de um novo vetor para o operador de cruzamento subsequente.

Zhang *et al.* (2021) desenvolveram o algoritmo Evolução Diferencial Binária Adaptativo no qual um operador de mutação é proposto e depende de uma função binária que utiliza um parâmetro adaptativo para definir o resultado em 0 ou 1.

CONCLUSÃO

A partir da análise dos artigos selecionados foi possível perceber que as técnicas utilizadas para adaptar a ED a espaços de busca binários utilizam 5 grupos de mecanismos: uso de operadores lógicos, utilização de funções de transferência, utilização de operadores de estimativa de probabilidade, meta-heurísticas combinadas e operadores de mutação não-paramétricos.

Também foi possível perceber que após a criação do algoritmo da ED em 1995, os autores que estudaram as variantes binárias do método tinham maior preocupação em criar mecanismo inovadores de binarização. Contudo, os trabalhos mais recentes têm maior preocupação em melhorar a performance do método, muitas vezes recorrendo à combinação de mecanismos previamente desenvolvidos por outros autores, podendo fazer uso inclusive de mecanismos que pertencem originalmente a outros métodos, conforme percebido na categoria de meta-heurísticas combinadas, cujos métodos integram elementos da PSO à EDB.

Outra tendência observada durante a análise dos artigos selecionados é a de desenvolver algoritmos específicos para determinadas aplicações. Isso parece acontecer à medida em que novas aplicações para a EDB vão sendo estudadas, pois os algoritmos existentes precisam ser adaptados para respeitar restrições inerentes aos problemas em estudo.

Quanto ao método de pesquisa, foi possível perceber que apenas 36% dos trabalhos encontrados se adequaram aos critérios de inclusão, a esse respeito sugerimos para trabalhos futuros que o termo de busca seja restringido a campos mais específicos como o título das publicações, para evitar resultados que apesar de conter as palavras-chave pesquisadas, utilizam o método de ED apenas como parâmetro de comparação com outros métodos ou em alguma aplicação sem modificar nenhum aspecto do mesmo. Uma busca assertiva reduz o tempo de análise dos resultados encontrados.

Sobre a aplicação da EDB, nenhum dos trabalhos analisados menciona algum aspecto relacionado à experiência do usuário. É evidente a preocupação da academia em produzir algoritmos de otimização sofisticados para diversas áreas de aplicação, porém existe uma lacuna no que tange ao processo de tornar esses métodos disponíveis para ampla utilização sem que seja necessário um intensivo treinamento do usuário final.

Por fim, espera-se que o presente estudo seja capaz de contribuir como facilitador para pesquisadores que desejam iniciar seus estudos na área de otimização fazendo uso de heurísticas, uma vez que existe uma ampla gama de métodos que se derivam em numerosas variantes o que pode ser complexo em primeiro momento. Neste sentido, os resultados aqui enumerados e categorizados podem servir como guia para quem deseje estudar a EDB como método de otimização.

REFERÊNCIAS

- ALI IM, ESSAM D, KASMARIK K. A novel differential evolution mapping technique for generic combinatorial optimization problems. **Applied Soft Computing**, **80**, 297-309, 2019.
- ALI IM, ESSAM D; KASMARIK K. Novel binary differential evolution algorithm for knapsack problems. **Information Sciences** **542**, 177-194, 2021.
- BANITALEBI A, AZIZ MIA, AZIZ ZA. A self-adaptive binary differential evolution algorithm for large scale binary optimization problems. **Information Sciences** **367-368**, 487-511. 2016;
- BIDGOLI AA, EBRAHIMPOUR-KOMLEH H; RAHNAMAYAN S. **A Novel Multi-objective Binary Differential Evolution Algorithm for Multi-label Feature Selection**. In: 2019 Congress on Evolutionary Computation(CEC). 2019. p. 1588-1595.
- CHEN Y, XIE W, ZOU X. A binary differential evolution algorithm learning from explored solutions. **Neurocomputing** **149**, 1038-1047, 2015.
- DENG C-S, ZHAO B-Y, DENG A-Y, LIANG C-Y. **Hybrid-coding Binary Differential Evolution Algorithm with Application to 0-1 Knapsack Problems**. In: 2008 International Conference on Computer Science and Software Engineering. 2008. p. 317-320.
- DENG C, ZHAO B, YANG Y, DENG A. **Novel Binary Differential Evolution Algorithm for Discrete Optimization**. In: 2009 Fifth International Conference on Natural Computation. 2009. p. 346-349.
- DENG C, ZHAO B, YANG Y, DENG A. **Novel binary Differential Evolution without scale factor F**. In: Third International Workshop on Advanced Computational Intelligence. 2010. p. 250-253.
- DHALIWAL JS, DHILLON JS. **A binary differential evolution based memetic algorithm to solve profit based unit commitment problem**. In: 2018 IEEE 8th Power India International Conference (PIICON). 2018. p. 1-6.
- DHALIWAL JS, DHILLON JS. Profit based unit commitment using memetic binary differential evolution algorithm. **Applied Soft Computing** **81**, 105502, 2019.
- DHALIWAL JS, DHILLON JS. Memetic binary differential evolution to solve wind-thermal profit based unit commitment problem. **Applied Soft Computing** **125**, 109105. 2022.
- FAN G-M, HUANG H-J. **A novel binary differential evolution algorithm for a class of fuzzy-stochastic resource allocation problems**. In: 2017 13th IEEE International Conference on Control & Automation (ICCA). 2017. p. 548-553.
- GAO T, LI H, GONG M, ZHANG M, QIAO W. Superpixel-based multiobjective change detection based on self-adaptive neighborhood-based binary differential evolution. **Expert. Syst. Appl.** **212**, 118811, 2023.
- GHOSH A, XUE B, ZHANG M. **Binary Differential Evolution based Feature Selection Method with Mutual Information for Imbalanced Classification Problems**. In: 2021 IEEE Congress on Evolutionary Computation (CEC). 2021. p. 794-801.
- HE X, HAN L. **A Novel Binary Differential Evolution Algorithm Based on Artificial Immune System**. In: 2007 IEEE Congress on Evolutionary Computation. 2007. p. 2267-72.
- HE Z, XIAO L, HUO Z, WANG T, WANG X. Fast Minimization of Fixed Polarity Reed-Muller Expressions. **IEEE Access** **7**, 24843-51, 2019.
- HSIEH F-S. **Comparison of Three Metaheuristic Algorithms for Optimization of Cyber Physical Systems**. In: 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC). 2021. p. 11-14.
- KRAUSE J, LOPES HS. **A Comparison of Differential Evolution Algorithm with Binary and Continuous Encoding for the MKP**. In: 2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence. 2013. p. 381-387.
- LI T, DONG H, SUN J. Binary Differential Evolution Based on Individual Entropy for Feature Subset Optimization. **IEEE Access** **7**, 24109-24121, 2019.
- LI T, DONG H. Unsupervised Feature Selection and Clustering Optimization Based on Improved Differential Evolution. **IEEE Access** **7**, 140438-140450, 2019.
- PARIJA SR, SAHU PK, SINGH SS. **Differential evolution for cost reduction in cellular network**. In: 2014 International Conference on High Performance Computing and Applications (ICHPCA). 2014. p. 1-5.
- SINGH DK, SRINIVAS K, DAS DB. **Dynamic channel assignment in TDMA mobile cellular systems using hybrid discrete binary differential evolution algorithm**. In: 7th Iberian Conference on Information Systems and Technologies (CISTI 2012). 2012. p. 1-7.
- STORN R, PRICE K. **Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces**. 1997. 341-59.
- TARKHANEH O, NGUYEN TT, MAZAHERI S. A novel wrapper-based feature subset selection method using modified binary differential evolution algorithm. **Inf. Sci. (Ny) [Internet]** **565**, 278-305, 2021.

- WANG L, FU X, MAO Y, ILYAS MENHAS M, FEI M. A novel modified binary differential evolution algorithm and its applications. **Neurocomputing** [Internet]. **98**, 55–75, 2012. Available from: <https://www.sciencedirect.com/science/article/pii/S0925231212004316>
- WANG P, XUE B, LIANG J, ZHANG M. Feature Selection Using Diversity-Based Multi-objective Binary Differential Evolution. **Inf. Sci. (Ny)** [Internet] **626**, 586–606, 2023. Available from: <https://www.sciencedirect.com/science/article/pii/S0020025522016127>
- WANG X, GUO P. **A novel binary adaptive differential evolution algorithm for Bayesian Network learning**. In: 2012 8th International Conference on Natural Computation. 2012. p. 608–612.
- WEI W, XUAN M, LI L, LIN Q, MING Z, COELLO COELLO CA. Multiobjective optimization algorithm with dynamic operator selection for feature selection in high-dimensional classification. **Appl. Soft. Comput.** [Internet] **143**, 110360, 2023. Available from: <https://www.sciencedirect.com/science/article/pii/S1568494623003782>
- WU C-Y, NU K-S. **Art-enhanced modified binary differential evolution algorithm for optimization**. In: 2012 International Conference on Machine Learning and Cybernetics. 2012. p. 1838–43.
- WU C-Y, TSENG K-Y. **Engineering Optimization Using Modified Binary Differential Evolution Algorithm**. In: 2010 Third International Joint Conference on Computational Science and Optimization. 2010. p. 501–5.
- XIE W, CHI Y, WANG L, YU K, LI W. **MMBDE: A Two-stage Hybrid Feature Selection Method From Microarray Data**. In: 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). 2021. p. 2346–51.
- XUAN M, LI L, LIN Q, MING Z, WEI W. **A Modified Decomposition Based Multi-objective Optimization Algorithm for High Dimensional Feature Selection**. In: 2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS). 2021. p. 109–13.
- ZHANG L, LI H. **BPSOBDE: A Binary Version of Hybrid Heuristic Algorithm for Multidimensional Knapsack Problems**. In: 2017. 13th International Conference on Computational Intelligence and Security (CIS). 2017. p. 54–8.
- ZHANG Y, GONG D, GAO X, TIAN T, SUN X. Binary differential evolution with self-learning for multi-objective feature selection. **Inf. Sci. (Ny)** [Internet]. **507**, 67–85, 2020. Available from: <https://www.sciencedirect.com/science/article/pii/S0020025519307819>
- ZHANG Y, CHEN H, CHEN W, XU L, LI C, FENG Q. Near Infrared feature waveband selection for fishmeal quality assessment by frequency adaptive binary differential evolution. **Chemom. Intell. Lab. Syst.** [Internet] **217**, 104393, 2021. Available from: <https://www.sciencedirect.com/science/article/pii/S0169743921001611>