

Performance Evaluation of a Hybrid NSGA-III for Multi and Many-Objective Optimization in Real-World Problems

Paulo Lopes do Nascimento
Federal Center for Technological
Education of Minas Gerais
Belo Horizonte, Brazil
nascimento.paulo@aluno.cefetmg.br

Dênis E.C. Vargas
Department of Mathematics
Federal Center for Technological
Education of Minas Gerais
Belo Horizonte, Brazil
denis.vargas@cefetmg.br

Elizabeth F. Wanner
Department of Computer Engineering
Federal Center for Technological
Education of Minas Gerais
Belo Horizonte, Brazil
efwanner@cefetmg.br

Abstract—Several real-world problems can be modeled as optimization problems in which multiple conflicting objectives must be optimized simultaneously. Evolutionary algorithms (EA) are able to identify a set of non-dominated solutions (Pareto front) and are commonly used to solve these problems. Hybrid EAs that combine various optimization techniques can leverage the strengths of each method involved, enhancing their overall effectiveness. This study evaluates the performance of the well-known NSGA-III hybridized with Differential Evolution, Sine Cosine, and Arithmetic algorithms on optimization problems based on real-world applications with three, four and five objective functions, extracted from the CEC 2021 competition. Performance comparisons between the hybrid and original versions of NSGA-III were conducted using the IGD+ and HyperVolume indicators. Statistical analysis via the Wilcoxon test revealed significant improvements in NSGA-III performance when the hybrid version is considered.

Keywords—multi-objective optimization problems, evolutionary algorithms, reference points, real-world problems

I. INTRODUÇÃO

Algoritmos evolutivos (AEs) são métodos muito populares na literatura que resolvem problemas complexos de otimização, tais como problemas compostos por vários objetivos conflitantes que devem ser atendidos simultaneamente. Quando um problema de otimização possui duas ou três funções objetivo ele é chamado de Problema de Otimização Multiobjetivo (MOP, do inglês *Multiobjective Optimization Problem*). Quando um problema de otimização possui mais de três funções objetivo ele é chamado de Problema de Otimização com Muitos Objetivos (MaOP, do inglês *Many-objective Optimization Problem*) [1].

AEs desenvolvidos para resolver MOPs são chamados de *Multiobjective Evolutionary Algorithms* (MOEAs), enquanto aqueles voltados para resolver MaOPs são chamados de *Many-objective Evolutionary Algorithms* (MaOEAs) [2]. Para ambos, o objetivo é obter um conjunto de soluções não-dominadas chamado de frente de Pareto. Um algoritmo bem conhecido e dos mais populares da literatura é o *Non-dominated Sorting Genetic Algorithm III* (NSGA-III) [3], que tem se mostrado efetivo na resolução tanto de MOPs quanto de MaOPs. Dessa

forma, o NSGA-III pode ser considerado tanto um MOEA quanto um MaOEA.

Para um melhor aproveitamento das características de cada MOEA ou MaOEA, hibridizações têm se destacado na literatura. Por exemplo, [4] hibridizaram o *Multiobjective Evolutionary Algorithm based on Decomposition* (MOEA/D) [5] com Evolução Diferencial (ED) [6] e propuseram o MOEA/D-DE, que teve um bom desempenho em problemas com frentes de Pareto complicadas. Outro exemplo é o de [7], que hibridizou ED com uma técnica de penalização adaptativa para o tratamento de restrições e aplicou em MOPs, apresentando boa performance nesses problemas. [8] também usou esta mesma hibridização para resolver MOPs aplicados à problemas de otimização estrutural do mundo real, problemas nos quais a ED tem se destacado na literatura [9].

Este trabalho utilizou um algoritmo híbrido envolvendo o NSGA-III com a ED, o Sine Cosine Algorithm (SCA) [10] e o Algoritmo de Otimização Aritmética (AOA) [11] para resolver MOPs e MaOPs baseados em aplicações do mundo real. Experimentos computacionais comparando as versões híbrida e original do NSGA-III foram realizados com 9 MOPs e MaOPs extraídos da competição do CEC de 2021 [12]. Uma análise dos resultados obtidos foi realizada nos indicadores de desempenho *Inverted Generational Distance Plus* (IGD+) [13] e *HyperVolume* (HV) [14], além da aplicação do Teste de Wilcoxon para verificar a existência de diferenças estatisticamente significativas.

O restante do artigo é organizado da seguinte forma: a seção II apresenta uma fundamentação teórica do funcionamento dos algoritmos e problemas utilizados no trabalho. A seção III discorre sobre a metodologia utilizada nesse trabalho, descrevendo os indicadores de performance e o delineamento dos experimentos computacionais. A seção IV faz uma análise e discute os resultados obtidos, enquanto as conclusões finais e indicação de possíveis trabalhos futuros são mostradas na seção V.

II. FUNDAMENTAÇÃO TEÓRICA

A. Problema de Otimização Multiobjetivo (POM)

Problemas de otimização que contém mais um objetivo são conhecidos como problemas multiobjetivo (POM), onde o desejado consiste em achar um conjunto ideal que é avaliado por múltiplas funções que são conflitantes entre si. Normalmente devido ao conflito entre estas funções não é possível minimizar uma sem ocorrer alguma perda em outra função objetivo. Portanto, ao invés de tentar achar uma conjunto de valores ideais o método de resolução para esses tipos de problemas trata-se de encontrar um conjunto de soluções que são competitivas entre si, podendo avaliar qual solução utilizar posteriormente.

Matematicamente, um POM pode ser definido como

$$\begin{aligned} \min \quad & f(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_P(\mathbf{x})] \\ \text{s.a.} \quad & \mathbf{x} \in D \subseteq \mathbb{R}^n \end{aligned} \quad (1)$$

onde $f_p : \mathbb{R}^n \rightarrow \mathbb{R}$ com $p = 2, \dots, P$ são as funções-objetivo, \mathbb{R}^n é chamado de espaço das variáveis e \mathbb{R}^P de espaço dos objetivos. Um POM é dito sem restrições se $D = \mathbb{R}^n$ e com restrições se $D \neq \mathbb{R}^n$. Em um POM com restrições, D é chamado de conjunto viável e seus elementos são pontos viáveis. POMs com 2 e 3 funções objetivo são denominados *Multi-objective Optimization Problem* e POMs com mais de 3 funções objetivos são chamados de *Many-Objective Optimization Problems* (MaOPs).

Em POMs, o desejado é encontrar um conjunto de soluções que sejam competitivas entre si. Isso implica na definição de um método comparativo a fim de estabelecer um grau de dominância entre duas soluções. Considere dois vetores $\mathbf{u} = (u_1, \dots, u_k)$ e $\mathbf{v} = (v_1, \dots, v_k)$. Dizemos que \mathbf{u} domina \mathbf{v} (denota-se por $\mathbf{u} \preceq \mathbf{v}$) se, e somente se,

$$(u_i \leq v_i \forall i = 1, \dots, k) \wedge (\exists j \in \{1, \dots, k\} | u_j < v_j)$$

Se $\mathbf{u} \not\preceq \mathbf{v}$ e $\mathbf{v} \not\preceq \mathbf{u}$, dizemos que \mathbf{u} e \mathbf{v} são não-dominados entre si.

O conjunto Q definido matematicamente como

$$Q = \{\mathbf{x} \in D | \forall \mathbf{y} \in Q, (\mathbf{x} \neq \mathbf{y}) \implies \mathbf{x} \not\preceq \mathbf{y} \wedge \mathbf{y} \not\preceq \mathbf{x}\} \quad (2)$$

$$f(\mathbf{x}) \not\preceq f(\mathbf{y}) \wedge f(\mathbf{y}) \not\preceq f(\mathbf{x}) \wedge \nexists \mathbf{z} \in D | f(\mathbf{z}) \preceq f(\mathbf{x}) \quad (3)$$

é chamado de Soluções Ótimas de Pareto do POM (também conhecido por Conjunto Pareto-Ótimo) e o conjunto $FP = \{f(\mathbf{x}) \in \mathbb{R}^P, \forall \mathbf{x} \in Q\}$ é chamado de Frente (ou Fronteira) de Pareto do POM.

B. NSGA-III

O algoritmo é baseado no NSGA-II, mas com mudanças significativas em seu operador de seleção e desenvolvido com foco em POMs. Ele trabalha com uma população de soluções candidatas e usa o conceito de dominação para classificar as soluções em diferentes níveis de eficiência. Na estratégia de dominação a população é dividida em vários níveis não dominados, chamados "níveis de não-dominância". O objetivo é que os indivíduos em cada nível sejam incomparáveis entre

si e não sejam dominados/superados em todos os objetivos por nenhum outro indivíduo de mesmo nível.

O NSGA-III introduz uma abordagem de pontos de referência bem-distribuídos para ajudar a manter a diversidade entre os membros da população. Isso significa que, em vez de usar apenas a dominação e *crowding distance*, o NSGA-III usa uma série de pontos de referência para orientar a seleção e atualização adaptativa das soluções. O NSGA-III tem várias aplicações em POMs e é particularmente útil em MaOPs. Sua flexibilidade em lidar com problemas desse tipo o torna uma ferramenta poderosa em uma variedade maior de domínios.

Nessa etapa, todos os membros da população pertencentes aos níveis de não-dominância do primeiro nível até o nível l são agrupados em um conjunto chamado S_t . Se o tamanho do conjunto S_t for igual ao tamanho desejado da população final (indicado por N), não é necessário realizar mais operações. Neste caso, a próxima geração começa com o conjunto $P_{t+1} = S_t$. No entanto, se o tamanho do conjunto $|S_t| > N$, significa que o conjunto S_t já contém membros de $l - 1$ níveis de não-dominância. Nessa situação, a próxima geração, representada pelo conjunto P_{t+1} , é formada pela união dos conjuntos F_i dos níveis de 1 a $l - 1$, ou seja, $P_{t+1} = \bigcup_{i=1}^{l-1} F_i$. Os membros restantes, necessários para atingir o tamanho desejado da população (indicado por $K = N - |P_{t+1}|$), são escolhidos a partir do último nível de não-dominância F_l .

O objetivo dos pontos de referência é garantir a diversidade nas soluções obtidas pelo algoritmo. Em geral, usa-se a abordagem sistemática proposta por [15], que coloca os pontos em um hiperplano normalizado de dimensão $M - 1$, que é igualmente inclinado a todos os eixos objetivos e intercepta cada eixo no valor 1. O número total de pontos de referência H em um problema com M objetivos é dado por uma fórmula a seguir.

$$H = \binom{M + p - 1}{p} \quad (4)$$

No NSGA-III proposto, além de enfatizar soluções não dominadas, também é dada ênfase a membros da população que estão de alguma forma associados a cada um desses pontos de referência. Como os pontos de referência são amplamente distribuídos em todo o hiperplano normalizado, as soluções obtidas também são propensas a serem amplamente distribuídas na fronteira de Pareto-ótimo.

C. Evolução Diferencial (ED)

O algoritmo de Evolução Diferencial (ED) é uma técnica de otimização global baseada na evolução natural que tem sido amplamente utilizada para resolver problemas complexos em diversas áreas, como otimização de funções não lineares, aprendizado de máquina, engenharia e ciências computacionais. Esse algoritmo foi proposto por [6] e tem sido objeto de muitos estudos e aprimoramentos desde então.

A ED baseia-se no conceito de evolução, mais precisamente, ele simula a evolução de uma população de soluções candidatas ao longo de várias gerações, com o objetivo de melhorar a qualidade das soluções encontradas. A ideia fundamental

da ED é criar uma nova geração de indivíduos, combinando as características de três membros escolhidos aleatoriamente da população atual. O processo de criação da nova geração envolve operações únicas de mutação e cruzamento entre os indivíduos selecionados.

D. Algoritmo Seno Cosseno (SCA)

O Algoritmo Seno Cosseno (SCA, do inglês *Sine Cosine Algorithm*) é um método de otimização inspirado nos conceitos matemáticos das funções seno e cosseno, desenvolvido para resolver problemas de otimização contínua. O SCA foi proposto por [10] como uma alternativa eficiente e poderosa para abordar problemas complexos de otimização, principalmente em domínios contínuos.

O SCA é fundamentado nas propriedades das funções seno ($\sin(x)$) e cosseno ($\cos(x)$). Essas funções são periódicas e oscilam entre -1 e 1, sendo a função cosseno um deslocamento da função seno. A combinação dessas funções trigonométricas permite explorar o espaço de busca de maneira eficiente e balanceada.

O SCA também utiliza um mecanismo de busca baseado em uma população de soluções candidatas, representadas como vetores de parâmetros. Cada solução candidata é associada a um valor de aptidão, que reflete a qualidade da solução dentro do problema de otimização. O algoritmo procura minimizar ou maximizar essa função de aptidão ajustando os parâmetros das soluções candidatas. As etapas do SCA são descritas abaixo:

Inicialização: Uma população inicial de soluções candidatas é gerada aleatoriamente dentro do espaço de busca. Cada solução é representada como um vetor de parâmetros que definem uma posição no espaço.

Atualização das Soluções: A cada iteração do algoritmo, as soluções candidatas são atualizadas com base na fórmula abaixo, permitindo que as soluções se movam no espaço de busca de maneira suave e exploratória.

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times |r_3 P_i^t - X_i^t|, & r_4 < 0.5 \\ X_i^t + r_1 \times \cos(r_2) \times |r_3 P_i^t - X_i^t|, & r_4 \geq 0.5 \end{cases} \quad (5)$$

onde r_1, r_2, r_3 e r_4 são números aleatórios entre 0 e 1.

Avaliação de Aptidão: Após a atualização das soluções, cada uma delas tem sua aptidão avaliada de acordo com a função objetivo do problema. Ou seja, calcula-se o valor objetivo associado a cada solução candidata.

Seleção e Atualização: Com base nas avaliações de aptidão, as soluções são selecionadas para serem atualizadas na próxima iteração. Soluções com maior aptidão são mantidas.

E. Algoritmo de Otimização Aritmética

O Algoritmo de Otimização Aritmética (AOA, do inglês *Arithmetic Optimization Algorithm*), proposto por [11], é uma técnica que se baseia nas operações aritméticas básicas (adição, subtração, multiplicação e divisão) para otimizar uma variedade de problemas. O cerne dessa abordagem é a utilização inteligente dessas operações para manipular valores,

variáveis e parâmetros de um problema, a fim de atingir o estado otimizado. O AOA pode ser aplicado em problemas de otimização contínua, discreta e combinatória, tornando-o uma ferramenta versátil para resolver diferentes tipos de desafios. As fórmulas de cruzamento do AOA são:

Dado $k_{UL} = k_{UL}$

$x_{i,j}(C_{\text{Iter}}) + 1 =$

$$\begin{cases} \text{best}(x_j) \div (MOP + \epsilon) \times (k_{UL} \times \mu + LB_j), & r_2 < 0.5 \\ \text{best}(x_j) \times MOP \times (k_{UL} \times \mu + LB_j), & \text{otherwise} \end{cases}$$

$x_{i,j}(C_{\text{Iter}}) + 1 =$

$$\begin{cases} \text{best}(x_j) - MOP \times (k_{UL} \times \mu + LB_j), & r_3 < 0.5 \\ \text{best}(x_j) + MOP \times (k_{UL} \times \mu + LB_j), & \text{otherwise} \end{cases}$$

III. METODOLOGIA

A. Problemas Benchmark

Um suíte de problemas de teste tem como objetivo englobar um conjunto de problemas com características distintas. Um de seus usos é fornecer um jeito prático de testar a eficiência de AEs, pois a grande diversidade nos tipos de POMs gera uma maior confiança na adaptabilidade do AE. Foram adotados neste trabalho 9 POMs do mundo real obtidos no conjunto introduzido por [12]. Esses problemas possuem 3 ou mais objetivos.

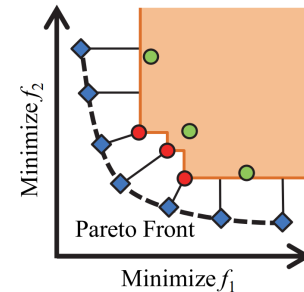
B. Medidas de Performance

Devido a natureza de POMs, não é uma tarefa intuitiva determinar o quão bem foi a performance do algoritmo. Contudo existem algumas medidas no qual analisam parte das características do algoritmo. Os indicadores de performance nos permitem, finalmente, analisar os impactos que mudanças no algoritmo tiveram em determinados problemas. Nesta pesquisa foi utilizado o IGD+ e o HyperVolume.

1) **IGD+:** Introduzido por [13], o IGD+ proposto é oficialmente definido como:

$$\text{IGD+}(A) = \frac{1}{|Z|} \left(\sum_{i=1}^{|Z|} d_i^{+2} \right)^{1/2} \quad (6)$$

Fig. 1. visualização do uso do IGD+

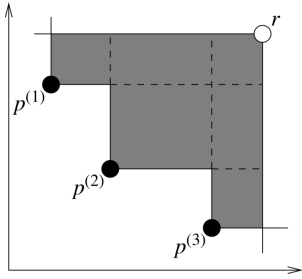


onde $d_i^+ = \max\{a_i - z_i, 0\}$ representa a distancia modificada de z_i para a soluão mais pr3xima de A , que corresponde ao valor a_i .

De modo simplificado, o c3lculo do IGD+ consiste em calcular a dist3ncia euclidiana entre o conjunto Z e A , onde para cada ponto de Z 3 selecionado o ponto mais pr3ximo que pertence ao conjunto A . O resultado do somat3rio 3 dividido pela quantidade de pontos do conjunto Z . Isso nos permite estabelecer uma dist3ncia m3dia entre o conjunto de solu3es atuais (A) e os pontos de refer3ncias (Z). O IGD+ requer como refer3ncia um conjunto de pontos n3o-dominados.

2) *HyperVolume*: Introduzido por [14], HyperVolume (HV) representa a medida de Lebesgue (3rea para o caso de POMs com dois objetivos e volume para o caso de POMs com tr3s objetivos) do espao dominado pelas solu3es obtidas em rela3o 3 um ponto de refer3ncia.

Fig. 2. visualiza3o do HyperVolume para duas fun3es objetivo



C. Experimentos Computacionais

Como mencionado, este trabalho utilizou um algoritmo h3brido (identificado neste texto como NSGA-III-HIB), onde uma combina3o dos AEs ED, SCA e AOA 3 incorporada ao NSGA-III para resolver POMs, incluindo MaOPs. A fim de comparar desempenhos do NSGA-III e do NSGA-III-HIB, executamos testes utilizando os 9 POMs do mundo real adotados. Esses problemas possuem restri3es, como pode ser verificado em [12]. Dessa forma, eles s3o conhecidos como RWCMPs (do ingl3s, *Real-world Constrained Multi-objective Optimization Problems*). Dos RWCMPs supracitados, os problemas com 3 objetivos s3o numerados em [12] como RWCMP8, RWCMP13, RWCMP17, RWCMP19, RWCMP41, RWCMP44 e RWCMP45, enquanto os MaOPs foram denominados RWCMP11 e RWCMP46. Cada algoritmo foi executado 25 vezes em cada problema, al3m de haver um requisito de parada sendo o n3mero m3ximo de avalia3es da fun33o (MaxFE), descrito como:

$$\text{MaxFE} = \begin{cases} 2.6250 \times 10^4, & \text{se } M = 3 \text{ e } D \leq 10 \\ 1.05 \times 10^5, & \text{se } M = 3 \text{ e } D > 10 \\ 1.43 \times 10^5, & \text{se } M = 4 \text{ e } D > 10 \\ 5.3 \times 10^4, & \text{se } M = 5 \text{ e } D \leq 10 \end{cases}$$

sendo M o n3mero de objetivos e D o n3mero de vari3veis de decis3o do problema em quest3o. Al3m disso, a popula3o utilizada por cada algoritmo foi de 105 para problemas com 3

fun33es objetivos, 143 para problemas com 4 fun33es objetivos e 212 para problemas com 5 fun33es objetivos.

A Tabela I apresenta o m3nimo, m3dia e m3ximo dos valores do IGD+ e HV obtidos pelos algoritmos NSGA-III e NSGA-III-HIB para os problemas RWCMPs juntamente com o n3mero m3ximo de gera3es utilizados em cada problema. A exist3ncia de diferenas estatisticamente significativas segundo o teste de Wilcoxon ($p\text{-valor} < 0,05$) 3 indicada na Tabela I pelo s3mbolo (+).

IV. AN3LISE E DISCUSS3O DOS RESULTADOS

Ap3s executar os algoritmos nos problemas, analisamos os valores do indicador IGD+, cuja regra 3 quanto menor o valor melhor o resultado, e do indicador HyperVolume, cuja regra 3 quanto maior o valor melhor o resultado. Al3m disso, 3 importante notar novamente que os valores com o s3mbolo (+) indicam que existem diferenas estatisticamente significativas entre os resultados, em rela3o ao melhor, de acordo com o teste de Wilcoxon. As Tabelas II e III resumem os resultados dos experimentos apresentados na Tabela I.

Entre os MaOPs, nominalmente os problemas RWCMP11 e 46, com 4 e 5 objetivos respectivamente, podemos concluir que o algoritmo NSGA-III-HIB apresentou o melhor resultado para o teste do IGD+ e do HyperVolume, uma vez que, al3m de ter obtido resultados melhores de maneira geral, tanto no IGD+ quanto no HV, houve diferena estatisticamente significativa pelo menos em um dos indicadores em cada problema (no problema 11 o IGD+ e no problema 46 o HyperVolume), al3m de o outro indicador tamb3m obter resultados melhores, mesmo sem diferena estatisticamente significativa.

Para os POMs com 3 objetivos, o NSGA-III-HIB obteve melhor resultado para HV em 4 dos 7 problemas, todos com diferenas estatisticamente significativas. Al3m disso, houve um empate no RWCMP13 em rela3o ao HV. Com rela3o ao IGD+, o NSGA-III-HIB obteve melhor resultado em 5 dos 7 problemas, 3 deles apresentando diferenas estatisticamente significativas.

Portanto, podemos concluir que o NSGA-III-HIB apresentou melhor desempenho global em rela3o ao NSGA-III para os POMs com origem no mundo real abordados nesse trabalho.

TABLE II
RESUMO DO RESULTADO DOS INDICATIVOS DO IGD+ DA TABELA I. VERDE INDICA QUE O NSGA-III-HIB OBTVEU O MELHOR RESULTADO E VERMELHO INDICA QUE O NSGA-III OBTVEU O MELHOR RESULTADO. O S3MBOLO (+) INDICA QUE EXISTE DIFERENA ESTATISTICAMENTE SIGNIFICATIVA.

Problema\Algoritmo	NSGA-III	NSGA-III-HIB
RWCMP08		
RWCMP11		(+)
RWCMP13		
RWCMP17		
RWCMP19		(+)
RWCMP41		(+)
RWCMP44		(+)
RWCMP45		
RWCMP46		

TABLE I
RESULTADOS DOS INDICADORES HYPERVOLUME (HV) E IGD+
DO ALGORITMO NSGA-III E DO NSGA-III-HIB NOS
PROBLEMAS RWCMPs.

Problema		NSGA-III	NSGA-III-HIB
RWCMP08	HV_max	0.0239	0.0238
	HV_média	0.0236	0.0237(+)
	HV_min	0.0234	0.0234
	HV_dp	1.2000e-04	8.9270e-05
	IGD+_max	0.1280	0.0830
	IGD+_média	0.0666	0.0612
	IGD+_min	0.0455	0.0486
	IGD+_dp	0.0170	0.0091
RWCMP11	HV_max	0.1080	0.1077
	HV_média	0.1060	0.1062
	HV_min	0.1046	0.1037
	HV_dp	9.1653e-04	9.7552e-04
	IGD+_max	3.4364e+04	3.3809e+04
	IGD+_média	2.6170e+04	2.3739e+04 (+)
	IGD+_min	1.9713e+04	1.8207e+04
	IGD+_dp	3.3093e+03	3.5400e+03
RWCMP13	HV_max	0.0437	0.0436
	HV_média	0.0435	0.0435
	HV_min	0.0433	0.0431
	HV_dp	1.1948e-04	1.6126e-04
	IGD+_max	10.0792	11.9448
	IGD+_média	7.0603	7.3815
	IGD+_min	5.4872	5.4909
	IGD+_dp	1.5582	1.9912
RWCMP17	HV_max	0.2560	0.2555
	HV_média	0.2435(+)	0.2255
	HV_min	0.2145	0.1353
	HV_dp	0.0108	0.0338
	IGD+_max	2.2750e+13	2.2748e+13
	IGD+_média	2.1692e+13	2.1091e+13
	IGD+_min	1.2070e+13	264.5495
	IGD+_dp	2.4858e+12	4.7301e+12
RWCMP19	HV_max	0.2055	0.2226
	HV_média	0.1842	0.2208(+)
	HV_min	0.1694	0.2190
	HV_dp	0.0063	8.0705e-04
	IGD+_max	4.6841e+03	314.1798
	IGD+_média	3.6907e+03	260.7722(+)
	IGD+_min	1.3447e+03	190.8370
	IGD+_dp	634.4033	36.7673
RWCMP41	HV_max	0.0987	0.0650
	HV_média	0.0420(+)	0.0298
	HV_min	0.0159	0.0205
	HV_dp	0.0202	0.0089
	IGD+_max	15.6629	12.2749
	IGD+_média	6.2516	2.0481(+)
	IGD+_min	1.0669	1.3136
	IGD+_dp	5.3094	2.1373
RWCMP44	HV_max	0.0018	0.0132
	HV_média	1.0593e-04	0.0073(+)
	HV_min	0.0000	0.0000
	HV_dp	3.8746e-04	0.0061
	IGD+_max	14.7915	9.6171
	IGD+_média	8.1229	3.1462(+)
	IGD+_min	1.2857	0.0166
	IGD+_dp	4.7769	4.3781

Problema		NSGA-III	NSGA-III-HIB
RWCMP45	HV_max	0.0777	0.0183
	HV_média	0.0033	0.0073(+)
	HV_min	0.0000	0.0000
	HV_dp	0.0155	0.0057
	IGD+_max	6.3987	5.4240
	IGD+_média	4.1447	3.5933
	IGD+_min	0.1638	2.5870
	IGD+_dp	1.6159	0.9674
RWCMP46	HV_max	0.0498	0.0508
	HV_média	0.0148	0.0220(+)
	HV_min	2.6534e-04	5.1090e-04
	HV_dp	0.0184	0.0178
	IGD+_max	11.6806	9.0756
	IGD+_média	6.8138	4.9383
	IGD+_min	0.6985	1.8214
	IGD+_dp	4.4380	2.9503

TABLE III
RESUMO DO RESULTADO DOS INDICATIVOS DO HYPERVOLUME DA
TABELA I. VERDE INDICA QUE O NSGA-III-HIB OBTVEU O MELHOR
RESULTADO, AMARELO INDICA QUE HOUVE EMPATE E VERMELHO INDICA
QUE O NSGA-III OBTVEU O MELHOR RESULTADO. O SÍMBOLO (+)
INDICA QUE EXISTE DIFERENÇA ESTATISTICAMENTE SIGNIFICATIVA.

Problema\Algoritmo	NSGA-III	NSGA-III-HIB
RWCMP08		(+)
RWCMP11		
RWCMP13		
RWCMP17	(+)	
RWCMP19		(+)
RWCMP41	(+)	
RWCMP44		(+)
RWCMP45		(+)
RWCMP46		(+)

V. CONCLUSÃO

Nesse trabalho foram analisados os desempenhos do NSGA-III-HIB, o qual é o NSGA-III hibridizado com a ED, SCA e AOA, bem como do NSGA-III originalmente proposto em 9 POMs com origem no mundo real, sendo 2 deles MaOPs. Devido ao fato desses problemas possuírem restrições, são conhecidos como RWCMPs, do inglês *Real-world Constrained Multi-objective Optimization Problems*. Uma análise estatística com os resultados obtidos foi realizada nos indicadores de desempenho IGD+ e HyperVolume, além do teste de hipóteses não-paramétrico de Wilcoxon para verificar a existência de diferenças estatisticamente significativas entre os resultados.

Com base nos testes realizados, é possível concluir que o NSGA-III-HIB apresentou melhor desempenho global em relação ao NSGA-III para os problemas com muitos objetivos abordados nesse trabalho, especialmente nos MaOPs. O NSGA-III-HIB obteve um desempenho superior em relação ao HV em 4 dos 7 POMs com 3 objetivos, todos eles com diferenças estatisticamente significativas. No POM RWCMP13 houve um empate no valor do HV. Além disso, o NSGA-III-HIB alcançou um melhor resultado em 5 dos 7 POMs com 3 objetivos, 3 deles evidenciando diferenças estatisticamente significativas, em relação ao IGD+.

Em resumo, este estudo mostrou que o NSGA-III-HIB é uma boa opção para POMs com origem no mundo real. Vale

ressaltar que não foram encontrados na literatura trabalhos que estendem variantes do SCA para lidar com MaOPs com origem no mundo real. Assim, tudo indica que esse trabalho foi pioneiro nesse quesito. Para trabalhos futuros, a análise do NSGA-III-HIB POMs com origem no mundo real com 2 funções objetivo poderá ser realizada para entender melhor o comportamento do NSGA-III-HIB.

AGRADECIMENTOS

Agradecemos ao CEFET-MG e à FAPEMIG (APQ 00408-21) pelo apoio concedido ao longo do projeto.

REFERENCES

- [1] Y. Tian, H. Wang, X. Zhang, and Y. Jin, "Effectiveness and efficiency of non-dominated sorting for evolutionary multi-and many-objective optimization," *Complex & Intelligent Systems*, vol. 3, pp. 247–263, 2017.
- [2] H. Chen, R. Cheng, J. Wen, H. Li, and J. Weng, "Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations," *Information Sciences*, vol. 509, pp. 457–469, 2020.
- [3] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints," *IEEE transactions on evolutionary computation*, vol. 18, no. 4, pp. 577–601, 2013.
- [4] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii," *IEEE transactions on evolutionary computation*, vol. 13, no. 2, pp. 284–302, 2008.
- [5] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [6] R. Storn and K. V. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997. [Online]. Available: <https://api.semanticscholar.org/CorpusID:5297867>
- [7] D. E. Vargas, A. C. Lemonge, H. J. Barbosa, and H. S. Bernardino, "Differential evolution with the adaptive penalty method for constrained multiobjective optimization," in *2013 IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 1342–1349.
- [8] D. Vargas, A. Lemonge, H. Barbosa, and H. Bernardino, "An algorithm based on differential evolution for structural multiobjective optimization problems with constraints (in portuguese)," *Rev Int Métodos Num Cálculo Diseño Ing*, vol. 32, no. 2, pp. 91–99, 2016.
- [9] J. P. G. Carvalho, D. E. Vargas, B. P. Jacob, B. S. Lima, P. H. Hallak, and A. C. Lemonge, "Multi-objective structural optimization for the automatic member grouping of truss structures using evolutionary algorithms," *Computers Structures*, vol. 292, p. 107230, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0045794923002602>
- [10] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-based Systems*, vol. 96, pp. 120–133, 2016.
- [11] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer methods in applied mechanics and engineering*, vol. 376, p. 113609, 2021.
- [12] A. Kumar, G. Wu, M. Z. Ali, Q. Luo, R. Mallipeddi, P. N. Suganthan, and S. Das, "A benchmark-suite of real-world constrained multi-objective optimization problems and some baseline results," *Swarm and Evolutionary Computation*, vol. 67, p. 100961, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2210650221001231>
- [13] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *Evolutionary Multi-Criterion Optimization*, A. Gaspar-Cunha, C. Henggeler Antunes, and C. C. Coello, Eds. Cham: Springer International Publishing, 2015, pp. 110–125.
- [14] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [15] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems," *SIAM J. on Optimization*, vol. 8, no. 3, pp. 631–657, Mar. 1998. [Online]. Available: <http://dx.doi.org/10.1137/S1052623496307510>