

# NSGA-III with Reference Points Adaptation

Rheidner A. C. Silva<sup>1</sup>, André Britto<sup>1</sup>

<sup>1</sup>Departamento de Computação – Universidade Federal de Sergipe (UFS)

rheidner.achiles@dcomp.ufs.br, andre@dcomp.ufs.br

**Abstract.** *Multi-Objective Evolutionary Algorithms (MOEAs) face certain difficulties solving problems with more than 3 objective functions, called Many-Objective Optimization Problems (MaOPs). New algorithms emerged to circumvent this problem, including NSGA-III. This algorithm explores the concept of reference points to make the selection of the solutions. But, it still has certain limitations and can be improved. This paper proposes two algorithms to adapt the reference points of the NSGA-III, based on the MOEA/D-AWA. The algorithms are evaluated to verify if the proposed adaptation procedure improves the performance of the NSGA-III.*

**Resumo.** *Algoritmos Evolucionários Multi-objetivo enfrentam dificuldades na resolução de problemas com mais de 3 objetivos, chamados Problemas de Otimização com Muitos Objetivos. Novos algoritmos então surgiram para contornar este problema, entre eles NSGA-III, que explora o conceito de pontos de referência para realizar a seleção de soluções. Porém, certas limitações ainda são apresentadas pelo algoritmo que pode ser melhorado. Assim, este trabalho propõe dois processos de adaptação dos pontos de referência no algoritmo NSGA-III, baseado no algoritmo MOEA/D-AWA. Os algoritmos com os dois processos propostos foram avaliados, buscando verificar se o processo de adaptação proposto melhora o desempenho do NSGA-III.*

## 1. Introdução

Um problema multiobjetivo (MOP, de MultiObjective Problem) apresenta mais de uma função objetivo para ser otimizada. Otimizar é, em síntese, achar um conjunto de soluções  $x$  capaz de minimizar (levando-se em consideração casos de minimização) tais funções objetivo. Normalmente essas diversas funções estão em conflito: diminuir o valor de uma função implica em aumentar o valor de outra e vice-versa. Portanto, não é possível achar uma solução única, que seja capaz de otimizar todas as  $m$  funções objetivo simultaneamente; em vez disso, busca-se o que se chama de soluções não-dominadas. Soluções não-dominadas são, resumidamente, soluções que possuem valor mínimo (considerando problemas de minimização) para alguma(s) das funções objetivo, dentre todas as soluções encontradas. A busca pelo conjunto dessas soluções é o alvo dos chamados MOEAs (Multi Objective Evolutionary Algorithms), tais como Algoritmos Genéticos, Otimização por Enxame de Partículas, Algoritmos Evolucionários, entre outros.

Nos últimos anos muitos MOEAs foram desenvolvidos e conseguiram chegar à bons resultados quando testados com MOPs de no máximo três funções objetivo. Porém, para problemas com um número maior de funções observa-se uma dificuldade em encontrar esse conjunto de soluções [Schutze et al. 2011]. Esse tipo de problema, com mais

de três funções a serem otimizadas, receberam então uma atenção especial e são por isso classificados como uma categoria diferente, os chamados problemas de muitos objetivos (MaOPs, de Many Objective Problems). Vários trabalhos também já foram realizados para a resolução desse tipo de problema; dentre eles, destaca-se a técnica do uso de pontos de referência nos MOEAs, presente em algoritmos recentes como o NSGA-III [Deb and Jain 2014] e o MOEA/D-AWA [Qi et al. 2014].

Essa utilização de pontos de referência conseguiu melhorar de forma significativa o funcionamento dos algoritmos em MaOPs, mas ainda apresentam problemas em algumas situações. Um exemplo são casos nos quais o conjunto de soluções não se encontram regularmente distribuídos pelo espaço, ou seja, casos onde não há distribuição uniforme das soluções. Isso se dá ao fato dos pontos de referência nos algoritmos propostos serem estáticos e fixos, não levando em consideração essa distribuição. Para contornar esse problema, algoritmos com pontos de referência adaptados foram sugeridos, mas ainda são poucos. Um exemplo deste tipo é sugerido em [Carvalho and Britto 2018], que ao adicionar três tipos de adaptação em seus pontos de referência, muda o algoritmo NSGA-III, obtendo, assim, bons resultados para MaOPs. Outro exemplo de tentar adaptar a funcionalidade do algoritmo à característica do problema foi proposto em [Qi et al. 2014] com o algoritmo MOEA/D-AWA, o qual adiciona uma adaptação aos vetores de peso do algoritmo MOEA/D, também obtendo bons resultados para MaOPs.

O objetivo deste artigo é propor dois processos de adaptação dos pontos de referência no algoritmo NSGA-III. Os métodos propostos usam de forma análoga o processo do MOEA/D-AWA para tentar adaptar os pontos de referência do algoritmo NSGA-III buscando uma melhora em seu funcionamento diante de MaOPs.

Para tal, dois algoritmos com processo análogos aos presentes no MOEA/D-AWA são propostos neste artigo, batizados como NSGA-III-ARPAI e NSGA-III-ARPAII. Ambos funcionam de forma semelhante, tendo diferenças apenas no que se baseiam para realizar a adaptação: o primeiro modifica o algoritmo do NSGA-III ao adicionar uma adaptação dos pontos de referência baseada nas distâncias entre os indivíduos de uma população; o segundo tenta realizar esta mesma adaptação baseado na distância entre os próprios pontos de referência.

Testes e comparações com outros algoritmos foram realizadas para poder verificar se as adaptações propostas melhoram ou pioram os resultados do algoritmo original NSGA-III, a partir da média e do desvio padrão de várias execuções dos mesmos, utilizando-se da medida de hipervolume e de testes estatísticos não paramétricos. Uma outra contribuição deste trabalho é a avaliação do desempenho dos métodos dos algoritmos analisados. Além de verificar o desempenho dos métodos propostos, este trabalho avalia o algoritmo MOEA/D-AWA em novos problemas, até então não explorados pelo algoritmo original, e também o compara ao algoritmo NSGA-III.

O resto deste trabalho organiza-se como o descrito à seguir: A Seção 2 descreve a fundamentação teórica necessária para compreensão básica dos algoritmos que são objetos de estudo deste trabalho. As propostas de adaptações dos pontos de referência são apresentadas na Seção 3. Resultados e comparações dos algoritmos com as adaptações e sem elas são discutidos na Seção 4. Por fim, a Seção 5 traz uma conclusão acerca do assunto proposto.

## 2. Fundamentação Teórica

Um MOP pode ser definido formalmente como (para problemas de minimização):

$$\text{Minimize } f(x) = (f_1(x), f_2(x), \dots, f_m(x)) \quad (1)$$

onde  $m$  é o número de funções objetivo. Portanto, o objetivo de uma otimização é achar o  $x$  que minimiza as várias funções  $f_i$ . Como dito anteriormente, as funções geralmente estão em conflito e, portanto, não é possível encontrar um único  $x$  que minimize todas as  $m$  funções ao mesmo tempo. O que é procurado então são as soluções não-dominadas.

Uma solução  $x_a$  somente domina uma solução  $x_b$  se

$$f_i(x_a) \leq f_i(x_b) \forall i \in \{1, 2, \dots, m\}. \quad (2)$$

Logo, uma solução não-dominada é aquela que não é dominada por nenhuma solução encontrada. Ao conjunto de soluções não-dominadas dá-se o nome de Conjunto Ótimo de Pareto (PS), e o conjunto das funções objetivo que essas soluções geram são chamadas de Fronteira de Pareto (PF). Encontrar o PS, e conseqüentemente o PF, é o objetivo principal de uma otimização em problemas multiobjetivo.

Problemas multiobjetivo com mais de três objetivos receberam uma designação especial por trazerem uma maior dificuldade aos algoritmos: são os chamados problemas de muitos objetivos (MaOPs). Vários algoritmos foram então criados com o objetivo de encontrar os tais conjuntos PS e PF para problemas com muitos objetivos. Dentre eles estão os algoritmos MOEA/D e NSGA-III, ambos bases dos algoritmos de estudo deste trabalho. A seguir são apresentados resumos dos funcionamentos destes dois algoritmos, como também do MOEA/D-AWA.

### 2.1. NSGA-III

Apresentado em [Deb and Jain 2014], o algoritmo NSGA-III busca garantir a diversidade das soluções com o uso dos pontos de referência. Tal algoritmo tem como base o NSGA-II [Deb et al. 2002], seu predecessor, diferindo deste ao adicionar o uso dos pontos de referência no processo de seleção das soluções. Tais pontos de referência encontram-se dispostos em um hiperplano de dimensão  $m$ , feito de acordo com o proposto em [Das and Dennis 1998]. O algoritmo então, faz o processo de seleção das soluções associando cada uma ao ponto de referência com a distância perpendicular mais próxima; caso duas delas se associem ao mesmo ponto de referência, é escolhido então aquele que apresenta a menor distância, assim assegurando a diversidade das soluções.

Os dois processos propostos neste artigo utilizam o NSGA-III como base. O algoritmo 1 mostra como uma geração é criada no NSGA-III, ou seja, como é gerada uma nova população  $P_{t+1}$  de indivíduos à partir de uma população anterior  $P_t$ . Em síntese, primeiro são realizados os processos normais presentes em MOEAs, como a recombinação e a mutação da população  $P_t$  gerando uma nova população  $Q_t$ .  $P_t$  e  $Q_t$  são então combinadas e o processo de organização por frentes característica dos algoritmos NSGAs começa. Frentes são apenas uma forma de organizar as soluções: soluções no frente  $F_1$  dominam as soluções no frente  $F_2$  e assim sucessivamente até o último frente. Para formar a nova população  $P_{t+1}$  são então considerados todas as soluções do frente  $F_1$ , depois do frente

$F_2$  até que se completem as  $N$  soluções necessárias em  $P_{t+1}$ . No último frente a ser inserido é feito, então, o processo de escolha baseado nos pontos de referência para que se possa escolher as  $K$  melhores soluções desse frente, onde  $K$  é o número de soluções restantes para que se complete  $N$ .

---

**Algoritmo 1** Geração  $t$  no NSGA-III

---

**Requer:** Estrutura de pontos de referência  $H$ .

Ponto  $z^*$ .

População pai  $P_t$ .

**Retorna:** Nova população  $P_{t+1}$

- 1:  $S_t = \emptyset, i = 1$
  - 2:  $Q_t = \text{Recombination} + \text{Mutation}(P_t)$
  - 3:  $R_t = P_t \cup Q_t$
  - 4:  $(F_1, F_2, \dots) = \text{Non-dominated-sort}(R_t)$
  - 5: **repita**
  - 6:      $S_t = S_t \cup F_i$  and  $i = i + 1$
  - 7: **até**  $|S_t| \geq N$
  - 8: Último frente para ser incluído  $F_l = F_i$
  - 9: **se**  $|S_t| == N$  **então**
  - 10:      $P_{t+1} = S_t$
  - 11: **senão**
  - 12:      $P_{t+1} = \bigcup_{j=1}^{l-1} F_j$
  - 13:     Pontos à serem escolhidos  $F_l : K = N - |P_t + 1|$
  - 14:     Normaliza objetivos e cria conjunto de referência  $Z^r$ :  
        $\text{Normalize}(f^n, S_t, Z^r, Z^s, Z^a)$
  - 15:     Associa cada membro  $s$  de  $S_t$  com um ponto de referência.
  - 16:     Calcula o nicho de cada ponto de referência.
  - 17:     Escolhe os  $k$  pontos restantes de  $F_l$  para completar  $P_{t+1}$  à partir do nicho dos pontos de referência.
  - 18: **fim se**
- 

## 2.2. MOEA/D e MOEA/D-AWA

A ideia central do MOEA/D, proposto em [Zhang and Li 2007], é transformar as várias funções objetivo em uma função escalar, através de um método de agregação de funções. Estes métodos funcionam atribuindo pesos a cada função e, a depender do método utilizam diferentes jeitos para transformar as diferentes funções objetivo em uma única função escalar, de acordo com os pesos anteriormente atribuídos. Após escolhida uma função de agregação, o algoritmo cria diferentes vetores de pesos e com estes criam diversas funções escalares diferentes, uma para cada vetor de peso. Então, o algoritmo começa o processo evolucionário, resolvendo simultaneamente as funções escalares criadas; cada uma delas irá gerar uma solução a cada geração; soluções não-dominadas encontradas são, então, guardadas.

Com o objetivo de tentar sanar alguns problemas verificados no algoritmo MOEA/D, tais como a não garantia da uniformidade das soluções encontradas, apesar da inicialização uniforme dos vetores e a dificuldade do algoritmo em lidar com PFs descontínuos e complexos, [Qi et al. 2014] propuseram o MOEA/D-AWA. O novo algoritmo propõe duas

mudanças chave no MOEA/D: uma nova inicialização dos vetores de pesos e a adição de um novo processo que tenta adaptar os vetores de pesos do algoritmo à forma do PF do problema. Tendo em vista o objetivo desta pesquisa, o processo de inicialização não foi levado em consideração, uma vez que o foco deste trabalho está na tentativa de uso análoga do processo de adaptação dos vetores feito no MOEA/D-AWA, nos pontos de referência do algoritmo NSGA-III.

O processo de adaptação proposto funciona de forma simples, mas eficaz. Primeiro uma quantidade  $nus$  (escolhida parametricamente) de vetores é removida, com base na sua distância de vizinhança, valor escolhido para representar a esparsidade de uma solução. A distância de vizinhança de um indivíduo (solução)  $j$  com relação a toda a população (conjunto de soluções atual) é calculada pela equação:

$$SL(ind^j, pop) = \prod_{i=1}^m L_2^{NN_i^j} \quad (3)$$

onde  $L_2^{NN_i^j}$ , representa a distância euclidiana do indivíduo  $j$  para o indivíduo  $i$ . À partir desta equação pode-se então concluir que indivíduos que estiverem em regiões povoadas terão valores de esparsidade pequenos e justamente por isso os  $nus$  subproblemas que escolhidos para serem removidos são os que possuem os indivíduos de menores esparsidades.

Após a remoção dos  $nus$  subproblemas, um segundo e último processo é feito para adicionar um número  $nus$  de novos subproblemas, que serão criados em regiões com pouca esparsidade de indivíduos. Esse processo leva em consideração o indivíduo com maior valor de distância de vizinhança, ou seja, o indivíduo que está na região mais esparsa do MOP. A seguinte equação, originada de dois teoremas demonstrados no artigo original, mostra como o MOEA/D-AWA obtém um novo vetor  $\lambda^{sp}$  a partir desse indivíduo:

$$\lambda^{sp} = \left( \frac{1}{\sum_{k=1}^m \frac{1}{f_k^{sp} - z_k^*}}, \dots, \frac{1}{\sum_{k=1}^m \frac{1}{f_m^{sp} - z_m^*}} \right), \prod_{j=1}^m (f_j^{sp} - z_j^*) \neq 0 \quad (4)$$

na qual  $z^*$  representa o ponto ideal, um dos parâmetros recebidos no MOEA/D original.

Todo o processo é feito periodicamente e somente inicia quando o algoritmo chega a uma determinada porcentagem do número de gerações total. Tanto o período quanto a porcentagem são escolhidos através de parâmetros passados ao algoritmo, que obteve bons resultados quando comparado com outros algoritmos já anteriormente propostos como o próprio MOEA/D e o NSGA-II em problemas da família ZDT e DTLZ.

### 3. NSGA-III-ARPA

Como já dito anteriormente este trabalho propõe a tentativa da utilização análoga do processo de adaptação dos vetores realizada no MOEA/D-AWA nos pontos de referência do algoritmo NSGA-III. Tal tentativa resultou em dois algoritmos diferentes, chamados posteriormente de NSGA-III-ARPAI e NSGA-III-ARPAII. A diferença primária entre os dois se dá no processo de remoção dos pontos de referência, no qual um se baseia na distância dos indivíduos e o outro se baseia na distância entre os próprios pontos de referência. A seguir são apresentados os processos de remoção e adição dos pontos de referência nos dois algoritmos propostos, além dos processos já integrados ao NSGA-III.

### 3.1. Algoritmos Gerais

Os algoritmos gerais NSGA-III-ARPAI e NSGA-III-ARPAII funcionam de acordo com o algoritmo básico do NSGA-III apenas adicionando o processo de adaptação dos pontos de referência de um dos dois propostos ao final do mesmo. Os dois processos somente são ativados depois de determinada porcentagem das gerações alcançadas e ocorre periodicamente a partir disso. Tanto o período quanto a porcentagem são definidos parametricamente. O algoritmo geral é apresentado em 2.

---

#### Algoritmo 2 Algoritmo Geral

---

**Requer:** Argumentos do algoritmo 1.

Número de gerações máxima  $t_{max}$ .

Porcentagem de gerações  $p$  de quando o processo de adaptações irá iniciar.

Número  $wag$  que indicará o tamanho do ciclo a partir do  $t_{max}$

**Retorna:** População ao final de  $t_{max}$  gerações.

- 1: **repita**
  - 2:     Algoritmo 1
  - 3:     **se** Número de geração atual  $t > p * t_{max}$  **então**
  - 4:         **se**  $t \bmod (t_{max}/wag) == 0$  **então**
  - 5:             removerARPAI( $P_t, p_{ref}, nus$ ) ou removerARPAII( $p_{ref}, nus$ )
  - 6:             adicionarPontos( $P_t, z^*, nus$ )
  - 7:         **fim se**
  - 8:     **fim se**
  - 9: **até** Número de gerações  $t_{max}$  ser alcançada
- 

### 3.2. Remoção de pontos de referência

O primeiro processo a ser realizado para a implementação do método proposto pelo algoritmo MOEA/D-AWA é a remoção de pontos de referência. O primeiro algoritmo proposto, NSGA-III-ARPAI, baseia-se na distância de vizinhança de um indivíduo para a remoção de um ponto de referência. Ou seja, indivíduos que estiverem em regiões povoadas do MOP têm o ponto de referência com a menor distância perpendicular removido. Por outro lado, no NSGA-III-ARPAII o critério de remoção é mudado, sendo baseado na distância entre os próprios pontos de referência, isto é: pontos de referência que estiverem em regiões povoadas são os escolhidos para serem removidos. Ambos os processos são descritos nos algoritmos 3 e 4.

### 3.3. Adicionando novos pontos de referência

Seguinte a remoção, vem o processo de adição de um novo ponto de referência. Tal adição se dá segundo a equação 4, que em ambos os algoritmos são baseadas no indivíduo com maior esparsidade. O processo é realizado segundo descrito no algoritmo 5. Os pontos de referência após um processo de adaptação (remoção e adição de pontos de referência) podem ser visualizados na figura 1.

## 4. Experimentação

Nesta seção serão apresentados como foram testados os algoritmos e como foram adquiridos os resultados apresentados.

---

**Algoritmo 3** removerARPAI

---

**Requer:** População atual *evolpop*.

Conjunto de pontos de referência *pref*.

Número de indivíduos removidos *nus*.

**Retorna:** Conjunto de pontos sem os *nus* pontos.

**Passo1.** Calcula a esparsidade (distância de vizinhança) de cada indivíduo na população atual seguindo a equação. 3.

**Passo2.** Deleta o ponto de referência que está à menor distância perpendicular do ponto de menor esparsidade:

**Se** Não atingiu o número *nus* de pontos removidos

**Então** Remove o ponto de referência que está à menor distância perpendicular do indivíduo de menor esparsidade. Vai para **Passo1**.

**Senão** Retorna *Pref* restante.

---

---

**Algoritmo 4** removerARPAII

---

**Requer:** Conjunto de pontos de referência *pref*.

Número de pontos removidos *nus*.

**Retorna:** Conjunto de pontos sem os *nus* pontos.

**Passo1.** Calcula a esparsidade (distância de vizinhança) de cada ponto de referência no conjunto de pontos de referência atual seguindo a equação. 3.

**Passo2.** Deleta o ponto de referência com menor esparsidade:

**Se** Não atingiu o número *nus* de pontos removidos

**Então** Remove o ponto de menor esparsidade. Vai para **Passo1**.

**Senão** Retorna *pref* restante.

---

---

**Algoritmo 5** Adição pontos de referência

---

**Requer:** População atual *evolpop*.

Ponto  $z^*$ .

Número de indivíduos removidos *nus*.

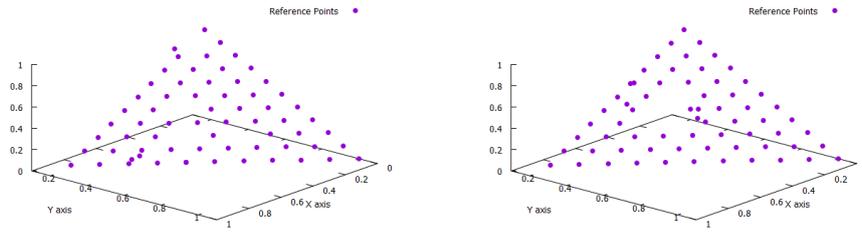
**Retorna:** População com os *nus* novos indivíduos.

**Passo1.** Calcula a esparsidade (distância de vizinhança) de cada indivíduo na população atual seguindo a equação. 3.

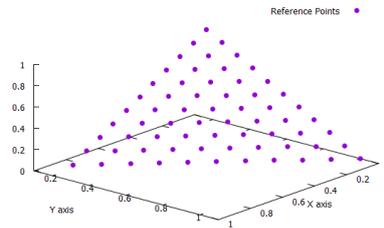
**Passo2.** Segue a equação 4 para a criação de um ponto de referência baseado no indivíduo de maior esparsidade.

**Passo3.** Caso o número *nus* de pontos tenha sido alcançado para-se o algoritmo.

---



(a) Pontos de referência após um processo de adaptação no algoritmo NSGA-III-ARPAI. (b) Pontos de referência após um processo de adaptação no algoritmo NSGA-III-ARPAII.



(c) Pontos de referência originais do NSGA-III.

**Figura 1. Comparação dos pontos de referência.**

#### 4.1. Metodologia

Os algoritmos MOEAD, MOEAD-AWA, NSGA-III, NSGA-III-ARPAI e NSGA-III-ARPAII foram testados seguindo a parametrização descrita a seguir. Seus resultados foram medidos utilizando-se da medida de hipervolume e então foram comparados segundo o testes estatístico de Wilcoxon. Com isso buscou-se analisar se a mudança de pontos de referência no NSGA-III, proposta neste artigo, melhora (ou piora) o desempenho do mesmo.

#### 4.2. Algoritmos e Parâmetros

O estudo realizado neste artigo busca analisar as ações das modificações propostas nos pontos de referência utilizados no NSGA-III. As duas modificações consistem em tentar adaptar o método AWA, utilizado por [Qi et al. 2014] nos vetores de pesos do MOEAD, para o uso análogo do mesmo nos pontos de referência do NSGA-III, cuja implementação foi obtida pelo framework jMetal.

Então, os cinco algoritmos alvos de discussão do estudo apresentados anteriormente foram testados e posteriormente comparados; são eles: MOEAD com a adaptação nos vetores de peso (MOEAD-AWA), NSGA-III com adaptação nos pontos de referência baseados na distância da população (NSGA-III-ARPAI), NSGA-III com adaptação nos pontos de referência baseados na distância dos próprios pontos de referência (NSGA-III-ARPAII), além dos próprios MOEAD e NSGA-III originais (ambos sem nenhuma modificação).

Os parâmetros de todos os algoritmos foram configurados como é relatado a seguir. Para o *crossover*, foi utilizado o SBX Crossover, com índice de distribuição do crossover em 30 e probabilidade 1. Para mutação, utilizou-se do Polynomial Mutation com os

valores  $n_m = 20$  para o índice de distribuição da mutação e  $p_m = 1/n$  para o valor de probabilidade de mutação. Para 3 objetivos o número de pontos de referência foi  $H = 91$ , para 8 objetivos este número foi de  $H = 120$  e para 10 objetivos  $H = 220$ . O tamanho da população de todos os algoritmos foram configurados com o mesmo valor do número de pontos de referência, descritos anteriormente. A condição de parada foi designada como alcançar o número máximo de avaliações da função objetivo, configurado em todos os algoritmos testados com o valor de 200000 avaliações.

Para os algoritmos que apresentam algum tipo de adaptação precisou-se definir os parâmetros *rateEvol*, *wag* e *rateAdapt* que representam a porcentagem a partir da qual começará o processo de adaptação, o tamanho do ciclo e a porcentagem de vetores ou pontos de referência que serão removidos/adicionados a cada processo de adaptação, o que gerará o número *nus*. Para todos os 3 algoritmos que apresentam adaptação esses parâmetros foram definidos como 0.8, 15 e 0.05, respectivamente, escolhidos através de testes. Todos os algoritmos foram executados 20 vezes para cada problema apresentado, levando em consideração 3, 8 e 10 objetivos.

### 4.3. Medidas e Testes estatísticos

Para avaliar os resultados obtidos a cada execução de cada algoritmo, utilizou-se da medida hipervolume. Tal medida foi escolhida por resumir em um único valor informações sobre a convergência do algoritmo em relação a Fronteira de Pareto verdadeira, como também da diversidade entre as soluções encontradas, além de ser uma das medidas mais utilizadas na literatura. Em resumo, seu funcionamento se dá com o cálculo do volume de todos os retângulos entre as soluções encontradas e um ponto extremo (*nadirpoint*,  $y^*$ ). Para os problemas DTLZ2 e CDTLZ2, o ponto  $y^*$  escolhido foi  $[2, 2, \dots, 2]$ ; para o problema DTLZ1  $y^* = [1.5, 1.5, \dots, 1.5]$ ; e para os problemas WFG1 e WFG3  $y^* = [3, 5, \dots, 2m+1]$ . Ao fim todos os resultados foram divididos por  $\prod_{i=1}^m y_i$ , a fim de normalizar os resultados em valores dentro do intervalo  $[0, 1]$ .

Para comparar os resultados obtidos com o teste, foram utilizados o teste estatístico não-paramétrico de Kruskal e o teste de post-hoc Nemenyi. O teste de Kruskal, permite a comparação de distribuição de diferentes amostras, tendo como resultado um valor, o qual indicará se alguma dessas amostras apresenta diferença estatisticamente significativa quando comparada à outra amostra. Após o teste de Kruskal o post-hoc Nemenyi foi então utilizado. Este serve para comparar as amostras duas a duas, retornando um valor *p-value* que indicará se uma possui diferença estatisticamente significativa em relação à outra. Neste trabalho considerou-se resultados estatisticamente diferentes aqueles que apresentaram *p-value*  $< 0.05$  quando comparados.

### 4.4. Resultados

A média e o desvio padrão do hipervolume desses resultados são apresentadas na tabela 1. Os símbolos ao lado dos resultados do NSGA-III-ARPAI e NSGA-III-ARPAII representam se o algoritmo foi melhor (+), pior (-) ou se apresentou resultado aproximado ( $\approx$ ) quando comparados com o algoritmo NSGA-III original. Os valores de *p-value* são apresentados nos gráficos da imagem 2. A partir dela é possível verificar que as adaptações propostas ao NSGA-III se igualam na maioria dos resultados, perdem em poucos e não ganham em nenhum dos casos quando comparadas ao próprio algoritmo NSGA-III, ape-

**Tabela 1. Resultados dos hipervolumes adquiridos.**

Problema	Objetivos	MOEA/D	MOEA/D-AWA	NSGA-III	NSGA-III-ARPAI	NSGA-III-ARPAII
DTLZ2	3	0,921 (0,000)	0,921 (0,000)	0,927 (0,000)	0,927 (0,000) ≈	0,927 (0,000) ≈
	8	0,953 (0,002)	0,970 (0,000)	0,977 (0,024)	0,987 (0,013) ≈	0,972 (0,027) ≈
	10	0,961 (0,001)	0,970 (0,001)	0,988 (0,009)	0,985 (0,010) ≈	0,983 (0,011) ≈
CDTLZ2	3	0,990 (0,002)	0,991 (0,000)	0,993 (0,000)	0,993 (0,000) ≈	0,993 (0,000) ≈
	8	1,000 (0,000)	1,000 (0,006)	1,000 (0,000)	0,987 (0,012) -	0,985 (0,012) -
	10	1,000 (0,000)	1,000 (0,001)	1,000 (0,000)	1,000 (0,000) ≈	1,000 (0,000) ≈
DTLZ1	3	0,990 (0,000)	0,990 (0,000)	0,992 (0,000)	0,992 (0,000) ≈	0,992 (0,000) ≈
	8	0,999 (0,000)	0,999 (0,000)	0,999 (0,000)	0,988 (0,011) ≈	0,995 (0,004) ≈
	10	0,999 (0,000)	0,999 (0,000)	0,999 (0,001)	0,999 (0,001) ≈	0,999 (0,001) ≈
WFG1	3	0,877 (0,002)	0,871 (0,003)	0,895 (0,009)	0,891 (0,008) ≈	0,890 (0,012) ≈
	8	0,855 (0,143)	0,878 (0,057)	0,739 (0,045)	0,755 (0,030) ≈	0,745 (0,037) ≈
	10	0,918 (0,008)	0,918 (0,011)	0,712 (0,025)	0,706 (0,030) ≈	0,712 (0,023) ≈
WFG3	3	0,712 (0,000)	0,711 (0,000)	0,708 (0,002)	0,710 (0,003) ≈	0,710 (0,003) ≈
	8	0,470 (0,004)	0,558 (0,016)	0,608 (0,020)	0,408 (0,092) -	0,546 (0,085) ≈
	10	0,522 (0,008)	0,544 (0,038)	0,596 (0,031)	0,596 (0,021) ≈	0,599 (0,018) ≈

sar do mesmo não acontecer com a adaptação do MOEA/D, que consegue ultrapassar o seu algoritmo original em alguns casos.

Outro ponto a atentar-se é o aumento da variedade de resultados que os algoritmos propostos apresentaram em alguns casos, fator observado pelo aumento no desvio padrão com relação ao algoritmo NSGA-III original. Isso deve ocorrer devido à forma como os algoritmos propostos tentam adaptar-se à cada situação em cada execução de cada problema, variando mais quando este assim o demanda, o que não acontece em algoritmos que contêm pontos estáticos.

Também é possível observar por meio desta tabela a dificuldade que os algoritmos apresentam ao lidar com fronteiras de Pareto mais complexas, que são os casos dos problemas da família WFG. É possível verificar que os algoritmos propostos chegam a passar a média do NSGA-III para alguns desses problemas, apesar de não apresentar resultado estatístico relevante. Em contrapartida, o pior resultado obtido com os algoritmos propostos se dá com o problema WFG3 com 8 objetivos, onde se constata a maior discrepância entre as médias de um dos algoritmos adaptados e do NSGA-III, que apesar de sofrer com o formato irregular de algumas fronteiras de Pareto, ainda apresenta resultados parecidos ou melhores quando comparados às suas versões adaptadas.

Em uma comparação geral, é observado certo equilíbrio nos resultados entre o NSGA-III, MOEA/D e MOEA/D-AWA. Para problemas da família DTLZ o algoritmo NSGA-III consegue melhores resultados; já para os problemas da família WFG o MOEA/D junto com o MOEA/D-AWA superam o NSGA-III algumas vezes. Observa-se que os dois primeiros obtiveram resultados parecidos, sendo que o MOEA/D-AWA, consegue superar sua versão para determinados problemas. É também possível notar o efeito que o aumento do número de objetivos causa em todos algoritmos, principalmente no problema WFG3, o qual possui a PF mais complexa.

Considerando todas essas análises é possível concluir que, apesar da melhora obtida com a versão adaptada MOEA/D-AWA constatada no artigo que a propõe e nos resultados aqui apresentados, o método de adaptação não se mostrou competitivo quando

aplicado ao NSGA-III. Portanto conclui-se que o método não se mostra interessante para MaOPS, quando utilizado no NSGA-III.

## 5. Conclusão e Trabalhos Futuros

Neste trabalho foram analisados os efeitos de duas adaptações dos pontos de referência no algoritmo NSGA-III. Baseado no trabalho que gerou o algoritmo MOEA/D-AWA, versão do MOEA/D com adaptação dos vetores de peso, dois algoritmos com processo análogos à essa adaptação para adaptar os pontos de referência do NSGA-III foram criados, implementados e testados. Os resultados dos testes mostraram que estas adaptações aplicadas ao NSGA-III não surtem o mesmo efeito de quando aplicada ao MOEA/D.

Trabalhos futuros podem investigar novos processos de adaptação, levando em consideração as diferentes transformações dos pontos de referências proposta por (Carvalho e Britto 2018). Além disso, um caminho é explorar métodos inteligentes de adaptação dos pontos de referência que podem ser baseados em técnicas de aprendizado de máquina.

## Agradecimentos

Este estudo foi financiado em parte pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de financiamento 001 e pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), projeto Universal número 425861/2016-3.

## Referências

- Carvalho, M. and Britto, A. (2018). Influence of reference points on a many-objective optimization algorithm. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 31–36.
- Das, I. and Dennis, J. (1998). Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657.
- Deb, K. and Jain, H. (2014). 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*, 18(4):577–601.
- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.
- Qi, Y., Ma, X., Liu, F., Jiao, L., Sun, J., and Wu, J. (2014). Moea/d with adaptive weight adjustment. *Evol. Comput.*, 22(2):231–264.
- Schutze, O., Lara, A., and Coello, C. A. C. (2011). On the influence of the number of objectives on the hardness of a multiobjective optimization problem. *IEEE Transactions on Evolutionary Computation*, 15(4):444–455.
- Zhang, Q. and Li, H. (2007). Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731.

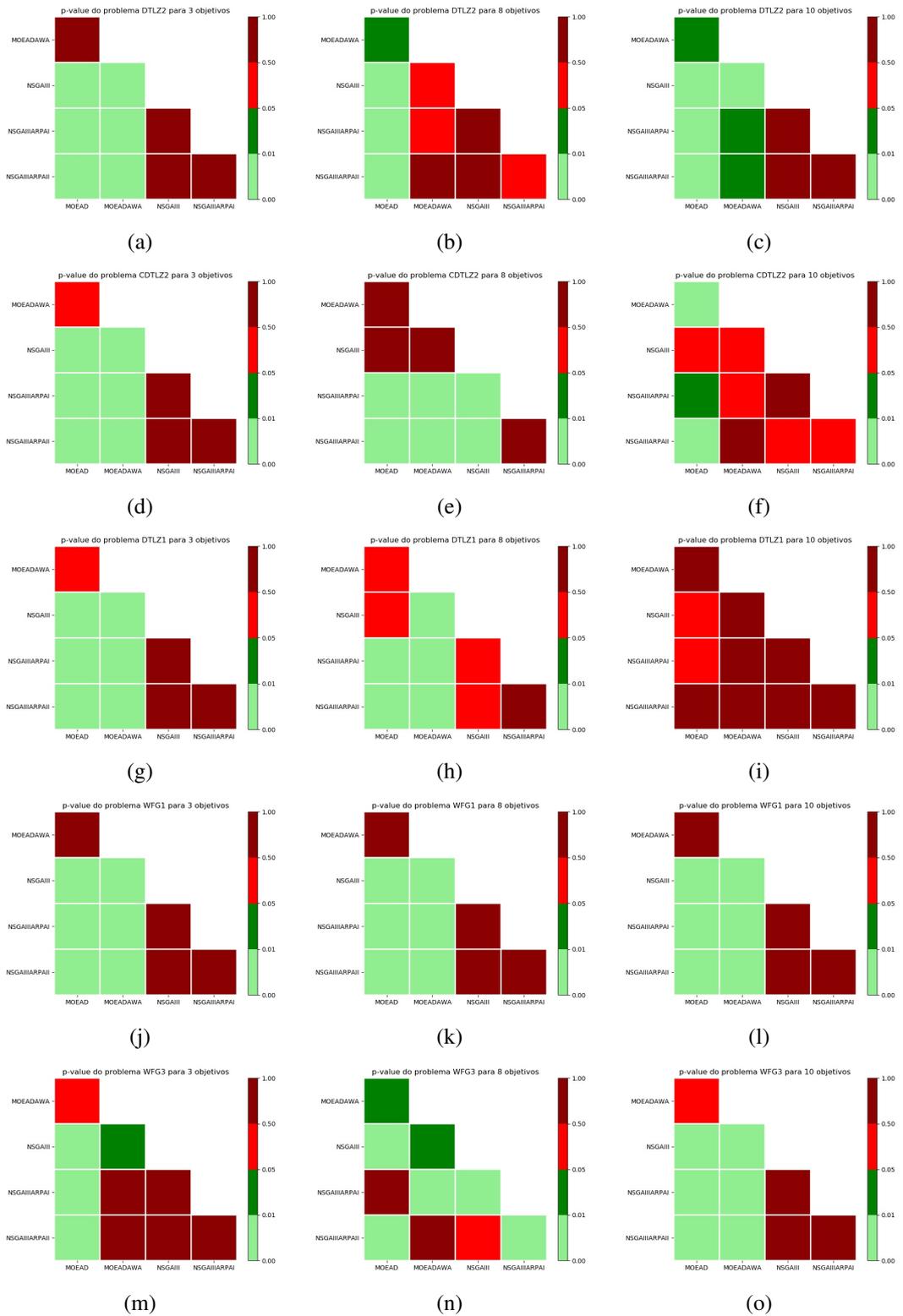


Figura 2. Comparação dos valores de  $p$ -value encontrados.