



## COMPARAÇÃO DE DESEMPENHO DOS ALGORITMOS “PSO” E “DE” NA OTIMIZAÇÃO DE UM RISER EM CONFIGURAÇÃO “LAZY - WAVE”

**Victor Q. Alves**

**Bruno F. Monteiro**

**Carl H. Albrecht**

victorqueiroz@poli.ufrj.br

brunomonteiro@lamcso.coppe.ufrj.br

carl@lamcso.coppe.ufrj.br

LAMCSO – Laboratory of Computational Methods and Offshore Systems,

PEC/COPPE/UFRJ – Civil Engineering Dept., Post-Graduate Institute of the Federal University of Rio de Janeiro.

Centro de Tecnologia, Bloco B sala B-101, CEP 21941-914 Rio de Janeiro, Brasil

**Resumo.** Risers de produção offshore são uma parte fundamental de sistemas de exploração de petróleo em águas profundas e ultra profundas, já que permitem a conexão entre as instalações submersas e as unidades de produção. O projeto desse tipo de estrutura requer um balanço complexo entre fatores de segurança e custo, o que motiva o uso de técnicas de otimização. Nesse contexto, este trabalho analisa a performance dos algoritmos Enxame de Partículas (PSO) e Evolução Diferencial (DE) na otimização de um riser de produção offshore em configuração “Lazy - Wave”. Os algoritmos e seus melhoramentos são brevemente descritos e um estudo paramétrico é realizado para ajustar os coeficientes ao problema de otimização do riser. Finalmente, ambos os algoritmos são comparados quanto a velocidade de convergência e melhor solução encontrada. Os resultados mostram que o PSO apresenta melhor velocidade de convergência enquanto o DE encontra resultados ligeiramente melhores..

**Palavras-Chave:** PSO, DE, Algoritmos Evolutivos, Riser de Produção Offshore

## 1 INTRODUÇÃO

A exaustão da exploração de reservas em águas rasas, somadas a mudanças na economia e descobertas de novas reservas impulsionaram a exploração de petróleo em águas profundas e ultra-profundas. Nesse contexto, surgiu a necessidade do estudo de *risers* de produção *offshore*. O projeto de *risers* exige um equilíbrio entre fatores de custo e segurança que resultam em um problema formal de otimização e síntese. De fato, a aceitação de uma determinada configuração depende de intensivas análises dinâmicas não-lineares, as quais demandam tempo devido ao elevado custo computacional. Com o objetivo de reduzir tal custo, diversos autores desenvolveram procedimentos que utilizam algoritmos evolutivos na otimização de *risers* de produção como em Vieira (2008), Tanaka (2006), Chen (2016) e Lima *et al* (2005). Além disso, algoritmos evolutivos também foram utilizados por Albrecht (2005) e Monteiro (2008) em procedimentos de otimização de linhas de ancoragem. Sendo assim, este trabalho é motivado pelas vantagens obtidas sobre problemas de otimização de *risers* ao se identificar características de desempenho de dois algoritmos evolutivos: Enxame de Partículas (PSO) e Evolução Diferencial (DE).

## 2 MODELAGEM DO PROBLEMA

O problema foi modelado conforme apresentado em Pina (2009). Ambos os algoritmos foram implementados no software *ProgOtim* desenvolvido no Laboratório de Métodos Computacionais e Sistemas Offshore – LAMCSO (COPPE-UFRJ). O programa permite dois tipos de análises. A primeira é baseada na análise dinâmica não-linear no domínio do tempo de Elementos Finitos (EF). Já a segunda se baseia num método de solução analítico de catenária, onde a forma do *riser* é gerada com o ajuste de três catenárias, sendo a intermediária uma catenária invertida que define a parte do *riser* que possui flutuadores. Embora esta última seja demasiadamente simples para aplicações reais, ela se mostra uma boa opção para comparar os algoritmos, dado o seu reduzido custo computacional.

### 2.1 Variáveis de Projeto

As variáveis de projeto presentes no problema de otimização estão apresentadas na Fig. 1.

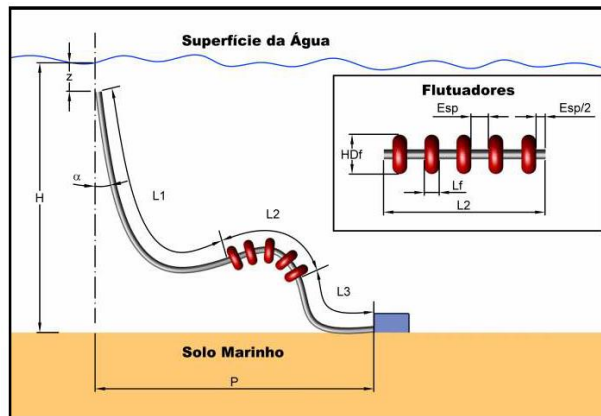


Figura 1: Variáveis de projeto no problema do riser em configuração lazy-wave.

Onde:

**L1:** Comprimento do segmento superior do riser;

**L2:** Comprimento do segmento com flutuadores distribuídos;

**L3:** Comprimento do segmento inferior do riser;

**α:** Ângulo de topo, isto é, o ângulo entre o riser e o eixo vertical na conexão com o sistema flutuante em equilíbrio;

**z:** Profundidade da conexão entre o riser e o sistema flutuante em equilíbrio;

**P:** Projeção do riser no plano horizontal;

**Lf:** Comprimento do flutuador;

**HDf:** Diâmetro do flutuador;

**Esp:** Espaçamento entre flutuadores.

Vieira (2008) mostrou que os parâmetros necessários para a análise se reduzem a: **L1, L2, L3, HDf, Lf, e Esp.**

## 2.2 Restrições de Projeto

As restrições de projeto incorporam os limites geométricos e estruturais do riser ao problema de otimização. Elas são verificadas através do método de solução analítico de catenária e estão listados a seguir:

- A tensão equivalente de *Von Mises* não deve superar a tensão permitida para garantir que o riser opere no regime elástico de deformação;
- O ângulo máximo entre o riser e o eixo vertical da conexão com a plataforma não deve ultrapassar um valor de segurança;
- Há uma variação máxima permitida do ângulo de *built-in*, medida no topo do eixo do riser, entre as configurações de equilíbrio estático (sem cargas ambientais) e instantâneas;
- Tensão máxima no topo do riser;
- Tensão mínima na parte inferior do riser, que evite o encurvamento e colapso de uma seção.

Uma vez que as restrições são especificadas, pode-se optar por penalizar ou excluir qualquer solução candidata que as desrespeite. Neste trabalho foi definida uma função de penalidade para todas as cinco restrições:

$$P = \begin{cases} k \times (1 - x^3), & \text{se } x < 1 \\ 0, & \text{se } x \geq 1 \end{cases} \quad (1)$$

Onde:

- $x$  é a razão entre o limite da restrição e o valor calculado;
- $k$  é um fator para se admitir soluções que violem as restrições.

## 2.3 Função de Custo

No caso do *riser* abordado neste trabalho, a função de custo é descrita por:

$$f = \frac{f_{min}}{(\sum_{i=1}^n IC_i \times L_i) + (V_{flut} \times I_{flut})} \quad (2)$$

Onde:

- $f_{min}$  é o menor custo possível, utilizado para normalizar  $f$  no intervalo  $[0,1]$ ;
- $IC_i$  é o índice de custo associado ao segmento de *riser*  $i$ ;
- $L_i$  é o comprimento do segmento de *riser*  $i$ ;
- $V_{flut}$  é o volume do flutuador;
- $I_{flut}$  é o índice de custo associado ao volume do flutuador.

## 2.4 Fitness ou Função Objetivo

Deseja-se maximizar o seguinte *fitness* para o problema de otimização do *riser*:

$$F = 100 \times \left[ \frac{f}{1,0 + \sum P_j} \right] \quad (3)$$

Onde:

- $P_j$  corresponde a penalidade relativa a violação do  $j$ -ésimo critério de restrição de projeto;
- $f$  é a função de custo (Eq. (2));
- O fator 100 é usado como fator de escala.

## 3 ENXAME DE PARTÍCULAS

O conceito de otimização pelo Método de Enxame de Partículas (PSO, do inglês *Particle Swarm Optimization*) foi proposto por Kennedy e Eberhart em 1995. Inicialmente, o interesse era simular comportamentos sociais como visto em enxames, cardumes e bandos de pássaros (Kennedy & Eberhart, 1995). No entanto, não demorou até que seu potencial para aplicação em otimização fosse reconhecido.

### 3.1 Algoritmo Básico

O algoritmo PSO pode ser dividido em 3 etapas básicas: i) geração de uma população aleatória onde cada indivíduo possui uma posição e velocidade; ii) atualização das velocidades; e iii) atualização das posições; segundo Monteiro *et al* (2011).

Cada partícula é representada por três vetores de dimensão  $N$  igual ao espaço de busca. Eles são: a posição atual ( $\vec{x}_i$ ), a melhor posição já visitada ( $\vec{p}_i$ ) e a velocidade ( $\vec{v}_i$ ). Após cada iteração, os vetores  $\vec{x}_i$  e  $\vec{v}_i$  são atualizados e o mesmo ocorre com o vetor  $\vec{p}_i$  se tal posição for melhor que todas as posições visitadas anteriormente. Desse modo, a tendência é que o vetor  $\vec{p}_i$  seja melhorado progressivamente. A melhor posição visitada por qualquer uma das partículas também é armazenada em um vetor  $\vec{p}_g$  para futuras comparações (Pina, 2009).

O “voo” de uma partícula é expresso da seguinte forma:

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \times \Delta t \quad (4)$$

Onde:

- $\vec{x}_i(t+1)$  é a posição futura da partícula;
- $\vec{x}_i(t)$  é a posição atual da partícula;
- $\vec{v}_i(t+1)$  é a velocidade atualizada da partícula;
- $\Delta t$  é a variação temporal unitária;

Ou seja, a posição futura da partícula é dada pela soma dos vetores posição atual e velocidade. O algoritmo trabalha ajustando  $\vec{v}_i$  o qual pode ser encarado como um tamanho de passo (Poli et al., 2007). O ajuste ocorre conforme a seguinte equação (Kennedy & Eberhart, 1995), (Shi & Eberhart, 1998):

$$\vec{v}_i(t+1) = \frac{C_1}{\Delta t} * rnd(\vec{p}_g - \vec{x}_i(t)) + \frac{C_2}{\Delta t} * rnd(\vec{p}_i - \vec{x}_i(t)) + \omega[\vec{v}_i(t)] \quad (5)$$

Onde:

- $\vec{v}_i(t+1)$  é a velocidade atualizada;
- $C_1$  é o coeficiente para o controle da componente social;
- $C_2$  é o coeficiente para o controle da componente cognitiva;
- $rnd$  é um número aleatório entre 0 e 1, segundo uma distribuição uniforme;
- $(\vec{p}_g - \vec{x}_i(t))$  é o termo de aceleração pela distância em relação a melhor posição já visitada por qualquer partícula;
- $(\vec{p}_i - \vec{x}_i(t))$  é o termo de aceleração pela distância em relação a melhor posição já visitada pela própria partícula;
- $\omega$  é o parâmetro de peso de inércia;
- $\vec{v}_i(t)$  é velocidade a ser atualizada.

A velocidade máxima ( $\vec{v}_{max}$ ) possível é limitada com base nos limites inferiores ( $\vec{x}_{min}$ ) e superiores ( $\vec{x}_{max}$ ) que podem ser alcançados pela partícula no espaço de solução. Isso evita que a velocidade cresça indefinidamente (Eberhart & Kennedy, 1995). Desse modo:

$$\vec{v}_{max} = \vec{x}_{max} - \vec{x}_{min} \quad (6)$$

Assim, as etapas básicas de execução do PSO podem ser escritas da seguinte forma (Albrecht, 2005); (Pina, 2010).

**Início**

Inicializa a população aleatória inicial P (posição e velocidade)

Avalia todas as partículas

Atualiza  $\vec{p}_g$

**Repita**

Atualiza velocidades

Atualiza posições

Avalia indivíduos

Atualiza  $\vec{p}_g$  e  $\vec{p}_i$

Até que o critério de parada seja satisfeito

**Fim**

Diversas modificações foram feitas no algoritmo básico PSO ao longo dos últimos anos com o objetivo de melhorar seu desempenho. A seguir serão apresentadas as modificações utilizadas nesse trabalho através do software *ProgOtim* desenvolvido no Laboratório de Métodos Computacionais e Sistemas Offshore - LAMCSO.

### 3.2 Variação do Coeficiente de Inércia ( $\omega$ )

Com o objetivo de mudar a ênfase de busca global para local do PSO, pesquisadores propuseram um coeficiente de inércia variável. Eberhart e Shi (2000) sugeriram uma variação linear ao longo das iterações. Já Chatterjee e Siarry (2006) utilizaram uma variação não linear para a adaptação de  $\omega$ . Neste trabalho, optou-se pela seguinte versão alternativa:

$$\omega(t) = \omega_0 \left[ 1 - \frac{(t-1)^n}{N^n} \right] \quad (7)$$

Onde:

- $\omega_0$  é o peso de inércia inicial;
- $N$  é o número máximo de iterações;
- $t$  é a geração atual;
- $n$  é o expoente de não linearidade;

### 3.3 Variação dos parâmetros social ( $C_1$ ) e cognitivo ( $C_2$ )

Ratnaweera, Halgamuge e Watson (2004) estenderam a ideia de variação linear ao longo das iterações também para os parâmetros  $C_1$  e  $C_2$  para se beneficiar de uma busca global nas iterações iniciais e uma busca local nas finais:

$$C_1(t) = (C_{1fin} - C_{1ini}) \frac{t}{N} + C_{1ini} \quad (8)$$

$$C_2(t) = (C_{2fin} - C_{2ini}) \frac{t}{N} + C_{2ini} \quad (9)$$

Onde:

- $C_{iini}$  é o valor inicial de  $C_i$ ;
- $C_{ifin}$  é o valor final de  $C_i$ .

## 4 EVOLUÇÃO DIFERENCIAL

O algoritmo de Evolução Diferencial (DE, do inglês *Differential Evolution*) realiza uma otimização estocástica implementando processos naturais observados na evolução biológica como seleção por aptidão, mutação e cruzamento. O DE, proposto por Storn e Price (1995), opera de forma similar aos demais algoritmos evolutivos. Sendo assim, algumas etapas básicas também podem ser identificadas (Das & Suganthan, 2011): i) Inicialização dos vetores; ii) Mutação baseada entre diferenças entre os vetores; iii) Recombinação (*Crossover*) e iv) Seleção.

### 4.1 Inicialização dos vetores

O DE se inicia com uma população aleatória de NP indivíduos representados na forma de vetores, cada um sendo uma solução multidimensional para o problema de otimização. Cada vetor possui dimensão D e as variáveis que os compõem representam as características do problema a ser modelado. As gerações são denotadas por  $G=0, 1, \dots, G_{\max}$ . Como os vetores podem sofrer mudanças ao longo das gerações usa-se a seguinte notação para representá-los:

$$\vec{x}_{i,G} = [x_{1,i,G}, x_{2,i,G}, \dots, x_{D,i,G}] \quad (10)$$

Onde:

- $\vec{x}_{i,G}$  é o i-ésimo vetor da população da geração G.

As componentes de cada vetor representando um indivíduo podem ainda ser restringidas por limites inferiores e superiores, para representar aspectos físicos do problema (Paiva, 2011):

$$x_{j,i,G} = x_{j,min} + rnd_{j,i} \times (x_{j,max} - x_{j,min}) \quad (11)$$

Onde:

- $x_{j,i,G}$  é a componente  $j$  do indivíduo  $i$  na geração G;
- $x_{j,min}$  é o menor valor que pode ser assumido pela componente  $j$ ;
- $x_{j,max}$  é o maior valor que pode ser assumido pela componente  $j$ ;
- $rnd_{j,i}$  é um número aleatório entre 0 e 1, segundo uma distribuição uniforme.

### 4.2 Mutação com a Diferença Entre Vetores

O algoritmo DE básico utiliza três vetores para realizar a mutação (Das & Suganthan, 2011):

- Vetor Alvo: é o indivíduo da geração atual que será o pai para a próxima geração;
- Vetor Doador: é o indivíduo obtido pela operação de mutação por diferença;

- Vetor Teste: é o descendente da combinação entre os vetores Alvo e Doador.

O DE cria um Vetor Doador para cada indivíduo (Vetor Alvo) da população atual usando três outros vetores,  $\vec{x}_{r_1^i}$ ,  $\vec{x}_{r_2^i}$  e  $\vec{x}_{r_3^i}$ . Os índices  $r_1^i$ ,  $r_2^i$  e  $r_3^i$  são números inteiros mutuamente excludentes obtidos aleatoriamente do intervalo [1, NP]. Eles também são diferentes do vetor base  $i$ . Em seguida, a diferença entre quaisquer dois vetores dos três escolhidos é multiplicada por um fator de perturbação ( $F$ ), normalmente pertencente ao intervalo [0,4;1,0]. Essa diferença é somada ao terceiro vetor obtendo-se o Vetor Doador (Das & Suganthan, 2011):

$$\vec{V}_{i,G} = \vec{x}_{r_1^i,G} + F \times (\vec{x}_{r_2^i,G} - \vec{x}_{r_3^i,G}) \quad (12)$$

Onde:

- $\vec{V}_{i,G}$  é o  $i$ -ésimo Vetor Doador obtido de outros 3 vetores aleatórios na geração  $G$ ;
- $\vec{x}_{r_1^i,G}$  é o vetor aleatório 1 selecionado para gerar o  $i$ -ésimo Vetor Doador na geração  $G$ ;
- $\vec{x}_{r_2^i,G}$  é o vetor aleatório 2 selecionado para gerar o  $i$ -ésimo Vetor Doador na geração  $G$ ;
- $\vec{x}_{r_3^i,G}$  é o vetor aleatório 3 selecionado para gerar o  $i$ -ésimo Vetor Doador na geração  $G$ ;
- $F$  é o fator de perturbação, normalmente escolhido entre [0,4;1,0], ou seja, real e constante.

### 4.3 “Crossover”

Para aprimorar a diversidade da população, Storn e Price (1995) propuseram a operação de *crossover*. Tal processo é bem conhecido na biologia e consiste na troca de material genético entre dois cromossomos para aumentar a diversidade dos descendentes. O objetivo da operação é gerar o Vetor Teste ( $\vec{U}_{i,G}$ ) resultante da combinação entre os vetores Doador e Alvo.

$$\vec{U}_{i,G} = [u_{1,i,G}, u_{2,i,G}, \dots, u_{D,i,G}] \quad (13)$$

O *crossover* pode ocorrer principalmente de duas formas, exponencial ou binomial. No caso binomial, as correspondentes componentes dos vetores Doador e Alvo são analisadas e trocadas se  $Cr$  for maior que um número gerado aleatoriamente entre 0 e 1. Portanto, a operação de *crossover* depende fortemente de  $Cr$ , denominado taxa de crossover.  $Cr$  é um parâmetro a ser ajustado no algoritmo DE, assim como o fator de perturbação  $F$ . A comparação é feita da seguinte forma (Paiva, 2011):

$$u_{j,i,G} = v_{j,i,G} \text{ se } rnd(0,1) \leq Cr, \quad (14)$$

*Vetor Teste recebe a componente do Vetor Doador*

$$u_{j,i,G} = x_{j,i,G} \text{ se } rnd(0,1) \geq Cr, \quad (15)$$

Vetor Teste recebe a componente do Vetor Alvo

Onde:

- $rnd(0,1)$  é um valor aleatório entre 0 e 1, novo para cada componente analisada;
- $Cr$  é a taxa de *crossover*.

#### 4.4 Seleção

Por fim, a função objetivo é calculada para os Vetores Teste e Alvo para que o melhor seja selecionado e ocupe um lugar na próxima geração. O menos apto é eliminado já que o número de indivíduos deve ser constante. (Cheng & Hwang, 2002). A operação de seleção é feita da seguinte forma:

$$\vec{X}_{i,G+1} = \vec{U}_{i,G} \text{ se } f(\vec{X}_{i,G}) \geq f(\vec{U}_{i,G}) \quad (16)$$

$$\vec{X}_{i,G+1} = \vec{X}_{i,G} \text{ se } f(\vec{X}_{i,G}) \leq f(\vec{U}_{i,G}) \quad (17)$$

Onde:

- $f(\vec{X})$  é a função objetivo a ser otimizada (maximizada no exemplo acima).

### 5 ESTUDO DE CASO

Com o objetivo de realizar uma comparação de desempenho entre os algoritmos DE e PSO, efetuou-se uma análise paramétrica sistemática para encontrar uma combinação de parâmetros que retorne bons resultados da função objetivo. Essa análise foi feita em cima do problema definido na seção 2. Ela consistiu basicamente em executar experimentos nos quais variou-se um determinado parâmetro mantendo-se os demais constantes de modo a verificar seu impacto na qualidade dos resultados. O passo de variação adotado foi 0,1 e cada experimento foi rodado 30 vezes para garantir robustez estatística.

A qualidade de cada experimento foi medida com base na média dos *fitness* médios populacionais, calculada com base nas 30 rodadas (nesta seção, a média dos 30 *fitness* médios populacionais será referenciada somente como **fitness médio** por simplicidade). Também foi analisada a média dos desvios padrões calculados para cada uma das 30 rodadas (referenciada somente como **desvio padrão**).

O procedimento foi realizado para cada parâmetro individualmente, além disso, combinações propostas por outros autores também foram testadas. Os experimentos admitiram um número máximo de 50 gerações o qual assegurou a convergência de ambos os algoritmos.

## 5.1 Definição do Problema

Um estudo de caso foi proposto com o objetivo de comparar as melhores soluções e convergência dos algoritmos DE e PSO. A configuração de riser “Lazy Wave” será otimizada para uma profundidade de 1290 metros e uma projeção horizontal de 2000 metros. As propriedades físicas e geométricas do mesmo se encontram na Tabela 1 a seguir. A Tabela 2 apresenta os intervalos das seis variáveis livres e as restrições admitidas. O segmento com flutuadores possui uma razão de custo ( $C2/C1$ ) igual a 2, o que significa que seu custo supera em duas vezes o do riser convencional.

Tabela 1: Propriedades físicas e geométricas do Riser.

| MATERIAL               |                        | GEOMETRIA                     |              |
|------------------------|------------------------|-------------------------------|--------------|
| Densidade              | 7800 kg/m <sup>3</sup> | Espessura                     | 0.01985 m    |
| Peso Específico        | 77 kN/m <sup>3</sup>   | Diâmetro Externo              | 0.21908 m    |
| Limite de Escoamento   | 413 MPa                | Diâmetro Interno              | 0.18098 m    |
| Tensão Admissível      | 277 MPa                | Peso do Flutuador             | 0.162 ton/m  |
| Módulo de Elasticidade | 207800 MPa             | Flutuabilidade do Flutuador   | 0.3175 ton/m |
| Razão de Custo C2/C1   | 2.0                    | Diâmetro Externo do Flutuador | 0.568 m      |

Tabela 2: Intervalos das variáveis livre e as restrições impostas ao problema

| VARIÁVEIS LIVRES                    | MIN(m) | MAX(m) | RESTRIÇÕES                  | VALOR     |
|-------------------------------------|--------|--------|-----------------------------|-----------|
| Segmento de Riser (L1)              | 800    | 2000   | Tensão de Von Mises         | 415.5 MPa |
| Segmento de Riser (L2)              | 400    | 800    | Ângulo de Topo Máximo       | 18°       |
| Segmento de Riser (L3)              | 800    | 2000   | Ângulo de Topo Mínimo       | 5°        |
| Diâmetro do Flutuador (Hdf)         | 1.2    | 3      | Variação de ângulo Built-in | 5°        |
| Comprimento do Flutuador (Lf)       | 0.8    | 3      | Tração Máxima no Topo       | 1000 kN   |
| Espaçamento entre Flutuadores (Esp) | 0.5    | 3      | Tração Mínima               | 300 kN    |

## 5.2 Escolha da População

A escolha da população deve ser feita com base em um balanço entre custo computacional e qualidade das soluções. Além disso, para fins de comparação, o mesmo número de indivíduos deve ser utilizado nos experimentos com DE e PSO. Assim, experimentos foram realizados variando-se a população com incrementos de 10 ou 5 indivíduos para o PSO e DE com o intuito de identificar o ponto de convergência do *fitness* médio de ambos. Para isso, foram utilizadas combinações padrão, sendo [ $C1 = C2$  1,7 e  $\omega = 0,6$ ] para o PSO e [ $Cr=0,2$  e  $F = 0,8$ ] para o DE (Trelea, 2002) (Pedersen, 2010). O mesmo computador foi utilizado em ambos os casos para garantir igual capacidade de processamento entre os algoritmos. Os resultados estão representados na Fig. 2.

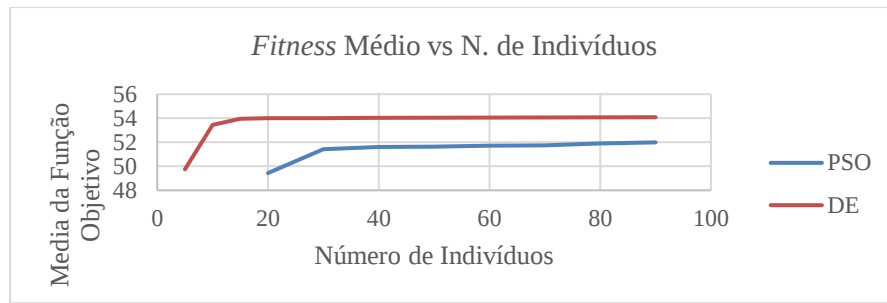


Figura 2: Fitness médio em função do número de indivíduos (PSO e DE).

Como se pode perceber da Fig. 2 o PSO atinge a convergência para 30 indivíduos enquanto o DE converge em torno de 15 indivíduos. Portanto a população escolhida para comparar os dois algoritmos foi de 30 indivíduos.

### 5.3 Análise paramétrica PSO

A qualidade dos resultados encontrados por algoritmos evolutivos depende fortemente da seleção dos parâmetros que controlam os algoritmos, motivando, portanto, uma análise paramétrica. Os parâmetros analisados foram apresentados na seção 2. O incremento utilizado na variação de todos os parâmetros foi de 0,1 enquanto um parâmetro era variado os demais eram mantidos constantes segundo uma combinação padrão sugerida por Trelea (2002):  $C1 = C2 = 1,7$  e  $\omega = 0,6$ .

$C1$  e  $C2$  variaram de 1 a 2; variou-se o coeficiente de inércia  $\omega$  de 0,3 a 1,2 para seu comportamento fixo;  $\omega_0$  variou de 0,3 a 1,2 para o caso linear e de 0,3 a 1,2 para o não linear. O expoente  $n$  utilizado na variação não linear de  $\omega$  também foi analisado para um intervalo de 0,6 a 1,8 (adotou-se um intervalo maior para esse parâmetro devido ao comportamento instável do *fitness* médio em função do mesmo). Por fim, testou-se combinações de  $C1$  e  $C2$  com pelo menos um deles variando linearmente:  $C1=1,7$  e  $C2$  de 1 a 2;  $C1$  de 1 a 2 e  $C2 = 1,7$ ;  $C1$  de 1 a 2 e  $C2$  de 2 a 1;  $C1$  de 2 a 1 e  $C2$  de 1 a 2;  $C1$  de 1 a 2,5 e  $C2$  de 2,5 a 1.

Os experimentos estão resumidos na Tabela 3 seguir, lembrando que foram admitidos uma população de 30 indivíduos, número máximo de 50 gerações e um incremento de 0,1 na variação dos parâmetros. Nela, o conjunto  $\{i \text{ a } j\}$  representa todas as combinações possíveis para o incremento de 0,1. Por exemplo:  $\{1 \text{ a } 2\} = \{1; 1,1; 1,2; 1,3; 1,4; 1,5; 1,6; 1,7; 1,8; 1,9; 2,0\}$ . Já a representação  $i : j$  (linear) significam que um único experimento foi feito, no qual os coeficientes  $C1$  e  $C2$  são variados linearmente de  $i$  a  $j$ .

Tabela 3: Resumo dos experimentos realizados com o PSO

| Experimento | C1                 | C2                 | $\omega$ (fixo) | $\omega_0$ (variação Linear) | $\omega_0$ (variação não linear) | n           |
|-------------|--------------------|--------------------|-----------------|------------------------------|----------------------------------|-------------|
| 1           | {1 a 2}            | 1,7                | 0,6             | n/a                          | n/a                              | n/a         |
| 2           | 1,7                | {1 a 2}            | 0,6             | n/a                          | n/a                              | n/a         |
| 3           | 1,7                | 1,7                | {0,3 a 1,2}     | n/a                          | n/a                              | n/a         |
| 4           | 1,7                | 1,7                | n/a             | {0,3 a 1,2}                  | n/a                              | n/a         |
| 5           | 1,7                | 1,7                | n/a             | n/a                          | {0,3 a 1,2}                      | 1,2         |
| 6           | 1,7                | 1,7                | n/a             | n/a                          | 0,6                              | {0,5 a 1,8} |
| 7           | 2 : 1 (linear)     | 1,7                | 0,6             | n/a                          | n/a                              | n/a         |
| 8           | 1,7                | 1 : 2 (linear)     | 0,6             | n/a                          | n/a                              | n/a         |
| 9           | 2 : 1 (linear)     | 1 : 2 (linear)     | 0,6             | n/a                          | n/a                              | n/a         |
| 10          | 1 : 2 (linear)     | 2 : 1 (linear)     | 0,6             | n/a                          | n/a                              | n/a         |
| 11          | 2,5 : 0,5 (linear) | 0,5 : 2,5 (linear) | 0,6             | n/a                          | n/a                              | n/a         |

Os resultados estão plotados a seguir. A Fig. 3 mostra a relação entre  $C1$  e  $C2$  (fixos), o *fitness* médio e o desvio padrão. A Fig. 4 demonstra a influência dos coeficientes de inercia fixo, linear e não linear sobre as médias da função objetivo e desvio padrão. O mesmo foi feito na Fig. 5 para o coeficiente  $n$ , utilizado na variação não linear do coeficiente de inércia. Finalmente, na Fig. 6 pode-se visualizar as combinações de  $C1$  e  $C2$  não lineares e seus respectivos *fitness* médio e desvio padrão.

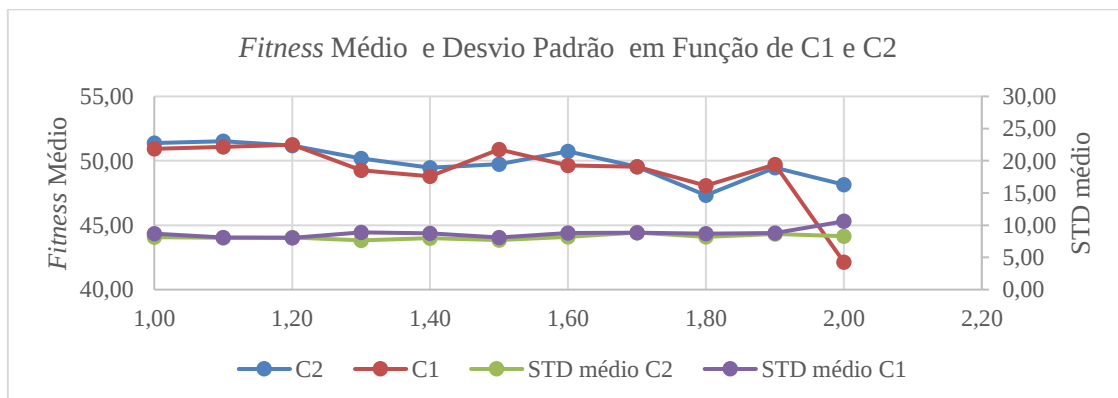


Figura 3: *Fitness* médio e desvio padrão em função de  $C1$  e  $C2$ .

Na Fig. 3 pode-se perceber um comportamento inversamente proporcional entre *fitness* médio e o desvio padrão. Este último apresentou leves variações em torno de 8. Além disso vemos uma tendência de diminuição *fitness* médio com o aumento dos coeficientes  $C1$  e  $C2$ , sendo que os maiores *fitness* médios obtidos ocorreram para os pontos [ $C1 = 1,2$ ; *fitness* médio = 51,249] e [ $C2 = 1,1$ ; *fitness* médio = 51,520].

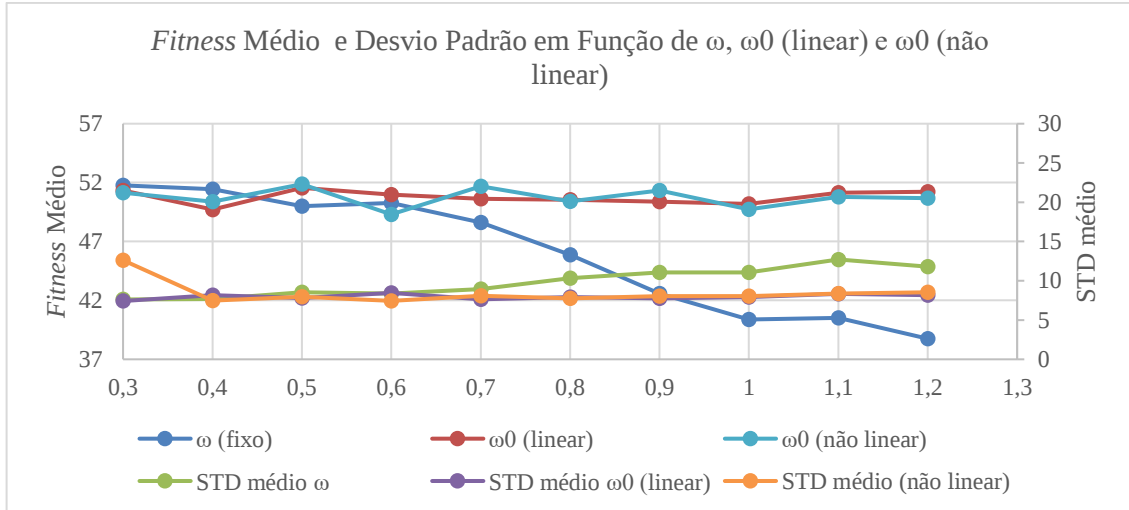


Figura 4: *Fitness Médio e Desvio Padrão em Função de  $\omega$ ,  $\omega_0$  (linear) e  $\omega_0$  (não linear).*

Da Fig. 4, pode-se inferir que o aumento do coeficiente de inércia ( $\omega$ ) tem um impacto negativo expressivo sobre o *fitness* médio. O mesmo não ocorre para  $\omega_0$  (linear) o qual demonstra uma melhora inicial em torno de 0,5, seguido por uma queda e recuperação somente em 1,1. Já  $\omega_0$  (não linear), apresentou uma oscilação no intervalo estudado em torno de um *fitness* médio igual a 51. O desvio padrão médio de  $\omega$  aumentou significativamente para maiores valores do coeficiente de inércia enquanto os demais se mantiveram quase constantes em torno de 7. Os maiores valores de *fitness* médio ocorreram para os pontos [ $\omega = 0,3$ ; *fitness* médio = 51,764], [ $\omega_0$  (linear) = 0,5; *fitness* médio = 51,555] e [ $\omega_0$  (não linear) = 0,5; *fitness* médio = 51,865].

Na Fig. 5 a seguir, pode-se visualizar o comportamento do *fitness* médio em função do coeficiente  $n$ .

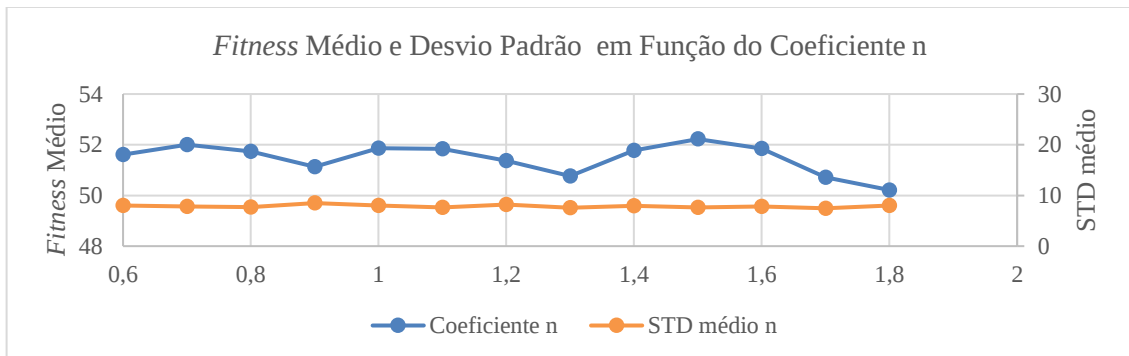


Figura 5: *Fitness Médio e Desvio Padrão em Função do Coeficiente  $n$ .*

Como mostra Fig. 5, o aumento coeficiente  $n$  tem um comportamento oscilatório sobre a Média da Função Objetivo, apresentando resultados de *fitness* médio que oscilam em torno de 51,5. O desvio padrão médio, entretanto, se manteve relativamente constante com o aumento de  $n$ . O maior *fitness* médio observado ocorreu no ponto [ $n = 1,5$ ; *fitness* médio = 52,223].

Na Fig. 6, estão representadas as combinações com  $C1$  e  $C2$  variando linearmente.

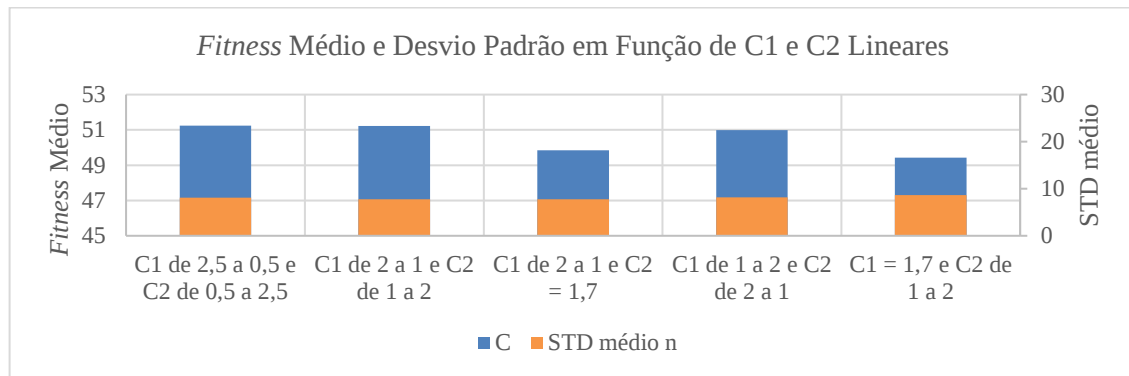


Figura 6: Fitness Médio e Desvio Padrão em Função de  $C1$  e  $C2$  Lineares.

Espera-se melhores resultados de *fitness* médio para combinações em que o  $C1$  inicia-se com grande valor e decai para valores menores, enquanto  $C2$  inicia-se pequeno. Tal comportamento favorece a busca global no início da execução do algoritmo e a busca local nas últimas gerações. Essa expectativa foi correspondida já que o melhor resultado ocorreu para a combinação de  $C1$  variando linearmente de 2,5 a 0,5 e  $C2$  de 0,5 a 2,5 a qual retornou um *fitness* médio de 51,234. O desvio padrão médio apresentou pequenas variações em torno de 8.

Com a análise paramétrica feita, foram realizados experimentos com as melhores combinações utilizando os melhores parâmetros encontrados. Elas estão resumidas na Tabela 4 abaixo e contemplam as 3 combinações com  $C1 = 2,5$  a 0,5 e  $C2 = 0,5$  a 2,5 variando linearmente e os coeficientes de inércia estático ( $\omega = 0,3$ ), linear ( $\omega_0 = 0,5$ ) e não linear ( $\omega_0 = 0,5$  com  $n = 1,5$ ). Analogamente, mais três combinações foram feitas para combinações de  $C1$  e  $C2$  estáticos e os três tipos de coeficiente de inércia. Finalmente, as combinações propostas por Trelea (2002) também foram testadas, elas são: [ $C1 = C2 = 1,7$  e  $\omega = 0,6$ ]; [ $C1 = C2 = 1,494$  e  $\omega = 0,729$ ].

Tabela 4: Resumo dos experimentos realizados com os melhores parâmetros obtidos na análise paramétrica para o PSO.

| Combinação | C1                 | C2                 | $\omega$ | Tipo de Variação de $\omega$ | Fitness Médio | Desvio Padrão |
|------------|--------------------|--------------------|----------|------------------------------|---------------|---------------|
| 1          | 2,5 : 0,5 (linear) | 0,5 : 2,5 (linear) | 0,3      | estático                     | 51,568        | 7,047         |
| 2          | 2,5 : 0,5 (linear) | 0,5 : 2,5 (linear) | 0,5      | linear                       | 49,908        | 7,1045        |
| 3          | 2,5 : 0,5 (linear) | 0,5 : 2,5 (linear) | 0,5      | não linear (n=1,5)           | 50            | 7,124         |
| 4          | 1,2                | 1,1                | 0,3      | estático                     | 49,795        | 7,181         |
| 5          | 1,2                | 1,1                | 0,5      | linear                       | 50,203        | 7,255         |
| 6          | 1,2                | 1,1                | 0,5      | não linear (n=1,5)           | 51,035        | 7,342         |
| 7          | 1,7                | 1,7                | 0,6      | estático                     | 50,272        | 8,906         |
| 8          | 1,494              | 1,494              | 0,729    | estático                     | 50,21         | 7,94          |

Como pode ser visto na Tabela 4, o maior *fitness* médio foi obtido para a combinação 1. Portanto, esta combinação será utilizada para comparação de

desempenho com uma combinação do algoritmo DE que passará por um processo similar de análise paramétrica.

## 5.4 Análise Paramétrica DE

A análise paramétrica para o algoritmo DE também foi realizada com um incremento de 0,1 para os coeficientes apresentados na seção 4.3. O intervalo analisado para o Fator de Perturbação ( $F$ ) foi [0,1 a 1] segundo Paiva (2011) e para a Taxa de Crossover ( $Cr$ ) foi [0,1 a 1] segundo Storn (1995).

Para uma população de 30 indivíduos e 50 gerações, foram realizados 4 tipos de variação. O primeiro com  $Cr$  variando de 0,1 a 1 e  $F$  constante igual a 0,8; o segundo com  $F$  variando de 0,1 a 1 e  $Cr$  constante igual a 0,2. Também foi feita uma análise com  $Cr$  e  $F$  variando juntos de 0,2 a 1 e, por fim, variando de maneira inversa da seguinte forma:  $Cr$  variando de 0,9 a 0,2 enquanto  $F$  varia de 0,1 a 0,8. As combinações dos experimentos estão resumidas na Tabela 5 a seguir.

Tabela 5: Resumo das combinações dos experimentos realizados para o DE.

| Tipo de variação          | Cr          | F           |
|---------------------------|-------------|-------------|
| Variação com F constante  | {0,1 a 1}   | 0,8         |
| Variação com Cr constante | 0,2         | {0,2 a 1}   |
| Variação Conjunta         | {0,2 a 1}   | {0,2 a 1}   |
| Variação Inversa          | {0,9 a 0,2} | {0,1 a 0,8} |

A seguir encontram-se os gráficos do *fitness* médio e do desvio padrão em função dos coeficientes  $Cr$  e  $F$  para as variações com  $Cr$  constante,  $F$  constante, ambos variando de forma conjunta e inversa.

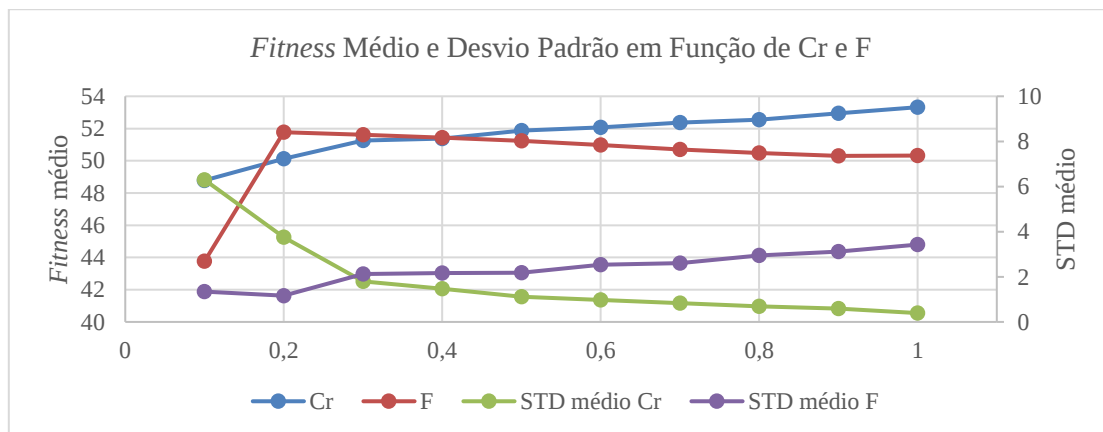


Figura 7: *Fitness* médio e Desvio Padrão em Função de  $Cr$  e  $CF$ .

Na Fig. 7 acima, demonstra a tendência do *fitness* médio em aumentar com o aumento de  $Cr$  ( $F$  mantido constante) e diminuir com o aumento de  $F$  ( $Cr$  mantido constante). O desvio padrão de  $Cr$  se comporta de maneira inversamente proporcional ao seu *fitness* médio e um comportamento similar é observado para o desvio padrão de  $F$ . Os melhores *fitness* médios ocorreram para os pontos [ $Cr = 1$  e  $F = 0,8$ ; *fitness*

médio = 53,328 ], no caso da análise de  $Cr$  e  $[Cr = 0,2$  e  $F = 0,2$ ;  $fitness$  médio = 51,770] no caso de  $F$ .

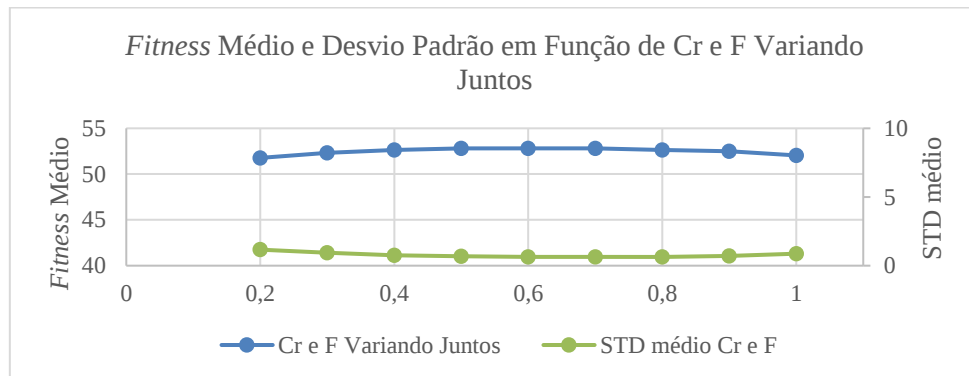


Figura 8: *Fitness Médio e desvio padrão em função de  $Cr$  e  $F$  Variando Juntos.*

A Fig. 8 mostra um leve aumento na média da função objetivo para  $Cr$  e  $F$  variando simultaneamente. O melhor resultado ocorreu para a combinação  $[Cr = F = 0,6$ ;  $fitness$  médio = 52,805], a partir desse ponto, a média da função objetivo tende a piorar com o aumento conjunto de  $Cr$  e  $F$ . Mais uma vez, o desvio padrão demonstra um comportamento inverso em relação ao  $fitness$  médio.

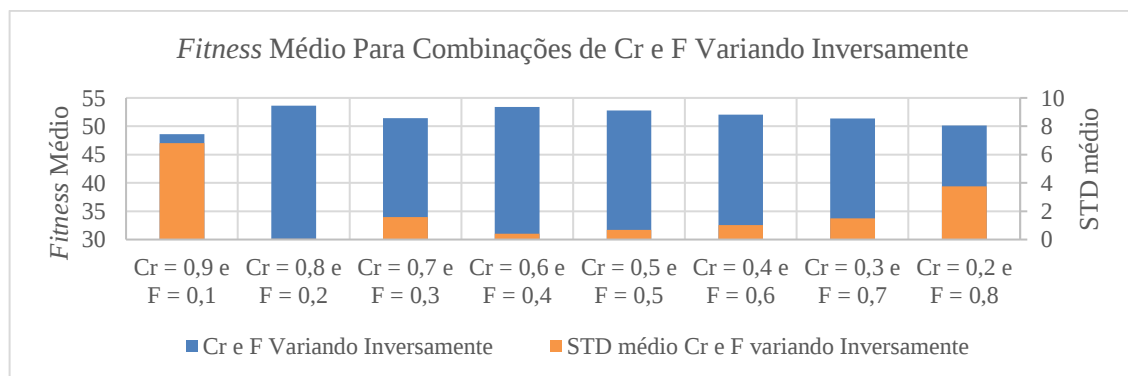


Figura 9: *Fitness médio e desvio padrão para combinações de  $Cr$  e  $F$  variando inversamente.*

A Fig. 9 mostra que pode-se obter bons resultados para valores de  $Cr$  e  $F$  próximos, na faixa de 0,4 a 0,6. No entanto, o melhor  $fitness$  médio ocorreu no ponto  $[Cr = 0,8$  e  $F = 0,2$ ;  $fitness$  médio = 53,651]. Os menores desvios padrão ocorreram para os maiores valores da média da função objetivo.

Após a análise paramétrica, foram realizados experimentos combinando os melhores parâmetros encontrados. Eles estão resumidos na Tabela 6 abaixo e totalizam 4 combinações. A combinação 1 utiliza os valores de  $Cr$  e  $F$  resultantes da análise feita com um parâmetro variando ( $Cr$  ou  $F$ ) enquanto o outro era mantido constante. A combinação 2 foi escolhida com base no tamanho da população e número de avaliações da função objetivo, segundo recomendado por Pedersen (2010). As combinações 3 e 4 resultaram da análise de  $Cr$  e  $F$  variando inversamente e juntos, respectivamente.

Tabela 6: Resumo das Melhores Combinações para o DE.

| Combinação | C     | F     | <i>Fitness</i> Médio | STD Médio |
|------------|-------|-------|----------------------|-----------|
| 1          | 1,000 | 0,200 | 52,336               | 0,950     |
| 2          | 0,943 | 0,661 | 51,642               | 1,094     |
| 3          | 0,800 | 0,200 | 53,651               | 0,040     |
| 4          | 0,500 | 0,500 | 52,808               | 0,688     |

Como pode ser visto na Tabela 6, a combinação 3 resultou no melhor *fitness* médio. Assim, essa combinação será utilizada para a comparação de desempenho em relação a combinação encontrada para o PSO.

## 5.5 Comparação de Desempenho entre DE e PSO

Utilizando as combinações resultantes da análise paramétrica, realizou-se uma comparação de velocidade de convergência entre DE e PSO aplicado a um caso de estudo de *riser* em configuração “lazy-wave”. Foram executadas 30 rodadas para fazer média com os maiores *fitness* registrado em cada uma das 50 gerações. Ou seja, foi possível acompanhar a evolução do melhor *fitness* médio encontrado pelos algoritmos ao longo das gerações, evidenciando qual convergiu primeiro e qual alcançou melhores valores de *fitness*. As combinações comparadas estão reiteradas na Tabela 7 e são apresentadas em conjunto com o *fitness* médio do melhor indivíduo na 50ª geração, seu desvio padrão, o *fitness* médio populacional (registrado na seção anterior) e o desvio padrão médio populacional. A evolução do *fitness* médio máximo está representada na Fig. 10 para ambos os algoritmos; assim como a evolução do menor valor da função objetivo e desvio padrão da média dos melhores valores. A Fig. 11 mostra a diferença percentual entre o *fitness* médio máximo do PSO e DE ao longo das 50 gerações.

Tabela 7: Combinações dos algoritmos PSO e DE escolhidos para comparação juntamente com os valores de *fitness* médio e desvio padrão.

| Algoritmo | Parâmetros                                                                  | <i>Fitness</i> Médio do Melhor Indivíduo | Desvio Padrão do Melhor Indivíduo | <i>Fitness</i> Médio Populacional | Desvio Padrão Médio Populacional |
|-----------|-----------------------------------------------------------------------------|------------------------------------------|-----------------------------------|-----------------------------------|----------------------------------|
| PSO       | C1 = 2,5 : 0,5 (linear); C2 = 0,5 : 2,5 (linear); $\omega = 0,3$ (estático) | 53,677                                   | 0,941                             | 51,568                            | 7,047                            |
| DE        | C = 0,8; F = 0,2                                                            | 53,73                                    | 0,94                              | 53,651                            | 0,04                             |

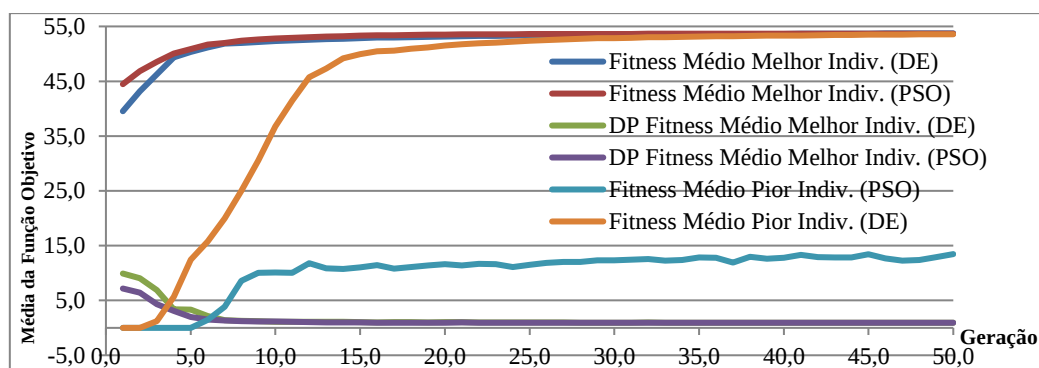


Figura 10: Evolução dos valores máximos e mínimos do fitness médio dos algoritmos DE e PSO e seus respectivos desvios padrões por geração.

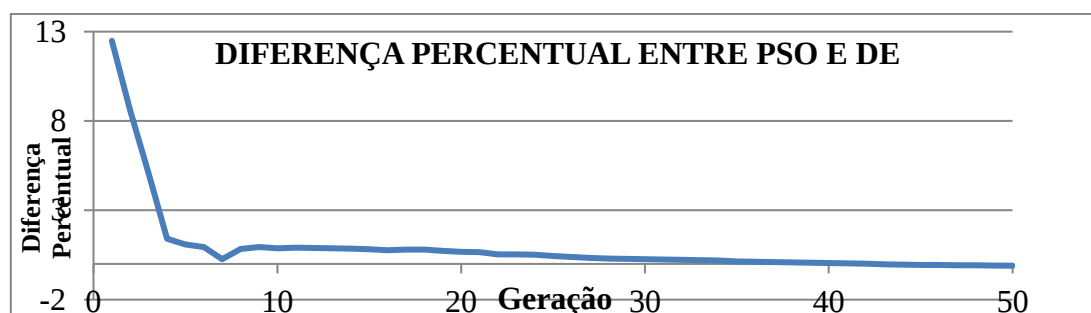


Figura 11: Diferença percentual entre o fitness médio máximo entre PSO e DE ao longo das 50 gerações.

A partir da Tabela 7 percebe-se que, embora o DE apresente um desvio padrão muito menor que o PSO para o *fitness* médio da população, o desvio padrão do *fitness* médio do melhor indivíduo é praticamente igual. Ou seja, a população do DE tende a se concentrar mais em torno do indivíduo de melhor *fitness* enquanto a população de PSO tende a estar mais espalhada pelo espaço de soluções. Além disso, a Tabela 7 mostra que o DE foi capaz de encontrar configurações ligeiramente melhores que o PSO ao fim das 50 gerações.

A Fig. 10 demonstra que o PSO possui maior capacidade de convergência quanto ao *fitness* do melhor indivíduo se comparado ao DE. Isso fica mais claro na Fig. 11, a qual apresenta a diferença percentual entre os *fitness* médios máximos do PSO e DE ao longo das 50 gerações (valores positivos indicam superioridade do PSO). Na primeira geração os melhores indivíduos produzidos pelo PSO apresentam *fitness* médio 12% maiores que os indivíduos do DE. Tal superioridade cai drasticamente até a quinta geração na qual o *fitness* médio do PSO é 1% maior que DE. Essa diferença decai levemente até que finalmente DE e PSO se equiparam na geração de número 40.

A Fig. 10 mostra ainda que os indivíduos de menor *fitness* médio do algoritmo DE se aproximam rapidamente dos indivíduos de maior *fitness* médio até a geração de número 15. Já os indivíduos de menor *fitness* médio do PSO melhoram somente até a décima geração, o que justifica seu elevado desvio padrão médio populacional.

## 6 COMENTÁRIOS FINAIS

Neste trabalho foram realizadas análises e comparações de desempenho entre dois algoritmos evolutivos, PSO e DE, na otimização de um *riser* em configuração “lazy-wave”.

Através de um estudo paramétrico, foi possível investigar a sensibilidade dos algoritmos em relação a cada um de seus parâmetros. No caso do PSO, percebe-se um maior impacto no *fitness* médio com a variação dos coeficientes social ( $C1$ ), cognitivo ( $C2$ ) e do coeficiente de inércia ( $\omega$ ) nos casos em que estes não variam ao longo das gerações. Ou seja, coeficientes que apresentam algum tipo de variação ao longo das gerações penalizam menos o *fitness* médio caso o intervalo de variação deles seja “mal selecionados”, se comparado aos coeficientes “estáticos”. A melhor combinação observada para o PSO foi [ $C1 = 2,5 : 0,5$  (linear);  $C2 = 0,5 : 2,5$  (linear) e  $\omega = 0,3$  (estático)].

No caso do DE, não foram consideradas variações nos parâmetros ao longo das gerações. Ainda assim, a taxa de *cross over* ( $Cr$ ) foi o parâmetro que apresentou maior impacto sobre o *fitness* médio se comparado ao fator de perturbação ( $F$ ).

A Figura 10 evidencia a superioridade quanto a velocidade de convergência do PSO se comparado ao DE. Já a Tabela 7 mostra que o DE é capaz de encontrar soluções um pouco melhores que o PSO. Vale ressaltar, que o processo de análise paramétrico do DE foi muito menos trabalhoso que o do PSO, já que este último contou com algumas alterações que aumentaram as possibilidades de análise paramétrica, mas que não melhoraram significativamente o *fitness* médio populacional.

## AGRADECIMENTOS

Agradeço à UFRJ e ao LAMCSO por possibilitarem a realização desse trabalho.

## REFERÊNCIAS

- Albrecht, C. H., 2005. *Algoritmos Evolutivos Aplicados à Síntese e Otimização de Sistemas de Ancoragem* (pp. 69-76), Rio de Janeiro: Universidade Federal do Rio de Janeiro.
- Bruno F. Monteiro, C. H. A. a. B. P. J., 2011. *An Enhanced Particle Swarm Optimization Method For The Design Of Oil Production Risers*. Coimbra, s.n., pp. 2-3.
- Chatterjee, A. & Siarry, P., 2006. *Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization*. s.l., s.n., p. 859–871.
- Cheng, S.-L. & Hwang, C., 2002. *Optimal approximation of linear systems by a differential evolution algorithm*. s.l., IEEE, pp. 698 - 707.
- Chen, J. et al., 2016. Flexible Riser Configuration Design for Extremely Shallow Water With Surrogate-Model-Based Optimization. *Journal of Offshore Mechanics and Arctic Engineering*, August.
- Das, S. & Suganthan, P. N., 2011. *Differential Evolution: A Survey of the State-of-the-Art*. s.l., IEEE, pp. 4-31.
- Eberhart, R. C. & Shi, Y., 2000. *Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization*. La Jolla, CA, USA, IEEE, pp. 84-88.
- Eberhart, R. & Kennedy, J., 1995. *A New Optimizer Using Particle Swarm Theory*. s.l., IEEE, pp. 39-43.

- EIA, U. E. I. A., 2016. Offshore production nearly 30% of global crude oil output in 2015. *TODAY IN ENERGY*.
- Kennedy, J. & Eberhart, R., 1995. *Particle Swarm Optimization*. s.l., IEEE, pp. 1942-1946.
- Lima, B. d. S. L. P. d., Jacob, B. P. & Ebecken, N. F. F., 2005. *A hybrid fuzzy/genetic algorithm for the design of offshore oil production risers*. s.l., s.n., pp. 1459-1482.
- Monteiro, B. d. F., 2008. *Aplicação do Método de Enxame de Partículas na Otimização de Sistemas de Ancoragem de Unidades Flutuantes para a Exploração de Petróleo Offshore*, Rio de Janeiro: COPPE/UFRJ.
- Paiva, A. L. d. O., 2011. *Aplicação do Método de Evolução Diferencial à Otimização de um Ciclo de Refrigeração por Compressão de Vapor de Dois Estágios Através da Análise Exérgica*, Rio de Janeiro: COPPE UFRJ.
- Pedersen, M. E. H., 2010. *Good Parameters for Differential Evolution*, s.l.: Hvass Laboratories.
- Pina, A. A. d., 2009. *Tailoring the particle swarm optimization algorithm for the design of offshore oil production risers*. s.l., Springer, pp. 5-7.
- Pina, A. A. d., 2010. *Metodologias de Análise, Síntese e Otimização de Sistemas Para a Produção de Petróleo Offshore Através de Metamodelos e Enxames de Partículas* (pp. 37-46), Rio de Janeiro: UFRJ/ COPPE.
- Poli, T. R., Kennedy, J. & Blackwell, T., 2007. *Particle swarm optimization An Overview*. s.l., Springer Science, pp. 33-57.
- Price, K. V., 2006. *Em: Differential Evolution - A Practical Approach to Global Optimization*. s.l.:Springer, pp. 37-134.
- Ratnaweera, A., Halgamuge, S. K. & Watson, H. C., 2004. *Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients*. s.l., IEEE, pp. 250-255.
- Shi, Y. & Eberhart, R., 1998. *A Modified Particle Swarm Optimizer*. s.l., IEEE, pp. 69-71.
- Storn, R., 1995. *On the Usage of Differential Evolution for Function Optimization*. Berkeley, s.n.
- Storn, R. & Price, K., 1995. *Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces*. Netherlands, Kluwer Academic Publishers, pp. 341 - 358.
- Tanaka, R. L. & Martins, C. d. A., 2006. *A Genetic Algorithm Approach to Steel Riser Optimization*. Hamburg, 25th International Conference on Offshore Mechanics and Arctic Engineering.
- Trelea, I. C., 2002. *The particle swarm optimization algorithm: convergence analysis and parameter selection*, s.l.: Elsevier Science.
- Vieira, L. T., 2008. *Otimização de Sistemas de Risers Para Exploração de Petróleo Offshore Através de Algoritmos Genéticos Paralelos*, Rio de Janeiro: COPPE/UFRJ.