ESTIMAÇÃO DE PARÂMETROS DE SISTEMAS NÃO LINEARES UTILIZANDO METAHEURÍSTICAS HÍBRIDAS PARALELIZADAS

Mariane Gavioli Bergamini[1], Jhonys Leite de Oliveira[1], Gustavo H. C. Oliveira[1], Gideon Villar Leandro[1]
[1]Programa de Pós-Graduação em Engenharia Elétrica (PPGEE), Universidade Federal do Paraná (UFPR), Curitiba, Brasil.
{gustavo, gede}@eletrica.ufpr.br
{marianegavioli, jhonys.oliveira}@ufpr.br

Resumo— Neste trabalho são propostos dois algoritmos híbridos de otimização, sendo cada um deles composto pela combinação das metaheuristicas: Algoritmo de Otimização de Leão (AOL), Evolução Diferencial (ED) e Busca Local Iterativa (BLI). Nestes algoritmos, as metaheurísticas são processadas de forma paralela e trocam informações, para melhorar seus respectivos desempenhos quando considerados individualmente. Adicionalmente, cada metaheurísta do algoritmo híbrido proposto contém um processamento paralelo visando redução de custo computacional. O contexto do presente artigo é o problema de estimação de parâmetros de modelos para sistemas não lineares. Portanto, os algoritmos propostos são comparados, para esta classe de aplicação, com versões clássicas das metaheurísticas citadas tomadas individualmente no que diz respeito à qualidade do resultado obtido e ao tempo de processamento. Através dos resultados obtidos, observa-se que o desvio padrão das soluções dos algoritmos híbridos propostos são menores que os métodos tomados individualmente. Neste mesmo cenário as técnicas de paralelismo apresentadas produziram uma redução de tempo computacional.

Palavra-chave: Algoritmo híbrido; evolução diferencial; algoritmo de otimização de leão; busca local iterativa; metaheurística paralela.

I. INTRODUÇÃO

Nas várias áreas da ciência e da engenharia, a solução de muitos problemas complexos passa por algum tipo de otimização não linear: controle de processos, escalonamento de tarefas, modelagem matemática, entre outras [1], [2].

Dentre estes problemas, destaca-se a identificação de sistemas cujo objetivo é representar o comportamento de sistemas reais através de modelos matemáticos [1], [3]. Na identificação de sistemas uma etapa de fundamental importância é a estimação dos parâmetros destes modelos [4], [5]. Frequentemente este é um problema de otimização não linear. Devido à dificuldade de se estimar parâmetros de modelos dinâmicos não lineares através de abordagens clássicas, vários algoritmos estocásticos baseados na genética, estratégia de evolução e programação evolutiva vêm ajudando a resolver este problema.

A motivação para utilizar esses algoritmos na estimação de parâmetros é devido ao seu alto desempenho ao lidar com problemas complexos em sistemas reais e obterem uma boa solução que com frequência converge para o ótimo global [6], [7].

As metaheurísticas [8] se beneficiam de operadores estocásticos, ou seja, tem abordagens não determinísticas onde a aleatoriedade é a principal característica desses algoritmos. Isso significa que eles operam de modo aleatório quando procuram a solução ótima global em espaços de busca. A aleatoriedade torna o algoritmo mais confiável na obtenção de solução a cada execução. Por ser um algoritmo que utiliza a formulação do problema para avaliar o conjunto de soluções, o processo é realizado independentemente do problema e com base nas entradas dadas e saídas recebidas [8].

Nas últimas décadas várias metaheurísticas têm sido propostas na literatura, cada uma apresentando vantagens e desvantagens nos vários problemas propostos. Porém, estas metaheurísticas possuem algumas semelhanças entre si, tais como: a reprodução, variação aleatória, competição e seleção do melhor indivíduo de uma população [9]. Entre os diversos tipos de algoritmos encontrados na literatura podem ser citados alguns: Algoritmo Genético [10],[11] ; Algoritmo Híbrido [12], [13]; Colônia de Formiga Leão[14], [15]; Programação Evolutiva [9], [16], [17]; Evolução Diferencial (ED) [7], [18], [19]; Partículas de Enxame [20],[21]; Busca Local Iterativa (BLI) [22], [23]; e Algoritmo de Otimização de Leão (AOL) [14], [24], [25].

Os algoritmos híbridos não seguem o paradigma de desenvolver uma única metaheurística tradicional. Eles combinam vários componentes das metaheurísticas provenientes de outras metaheurísticas de áreas diferentes da otimização [26]. São combinações de técnicas colaborativas e integradas onde ocorre troca de informações entre as metaheurísticas [27]. A principal motivação para se utilizar essa técnica é por proporcionar soluções mais eficientes levando a uma maior flexibilidade ao lidar com problemas de otimização em grande escala.

Neste trabalho são propostos dois algoritmos híbridos paralelos, sendo o primeiro algoritmo híbrido composto pela Evolução Diferencial (ED) e Algoritmo de Otimização de Leão (AOL); e o segundo algoritmo híbrido composto pelo AOL e Busca Local Iterativa (BLI). Nestes algoritmos híbridos, as metaheurísticas incorporadas trocam informações de forma colaborativa melhorando o desempenho da solução (robustez) do problema

em relação aos algoritmos citados tomados individualmente. Além disso, apresentam menor custo computacional para se chegar à solução.

Uma segunda contribuição deste trabalho está na descrição de um procedimento para tornar paralela a execução do algoritmo de otimização de leão.

Finalmente, todas as propostas são analisadas e comparadas dentro do contexto do problema de estimação de parâmetros de modelos de sistemas não lineares.

O presente artigo está organizado da seguinte maneira: na Seção 2 é descrita sobre a metodologia de estimação de parâmetros na identificação de sistemas, a Seção 3 é destinada aos algoritmos ED, AOL, BLI e a computação paralela nas metaheurísticas, na Seção 4 é descrito os algoritmos híbridos, a Seção 5 é dedicada às simulações e resultados obtidos e apresentação dos resultados obtidos. Finalmente, na Seção 6 são apresentadas as considerações finais do trabalho.

II. ESTIMAÇÃO DE PARÂMETROS NA IDENTIFICAÇÃO DE SISTEMAS NÃO LINEARES

Modelar sistemas dinâmicos não lineares é uma tarefa desafiadora [1], [28]. Ao contrário dos sistemas lineares, onde é possível visualizar famílias de modelos lineares, os sistemas não lineares podem assumir formulações diversas. Em [27], são discutidas estruturas dinâmicas para a representação de sistemas não lineares. A determinação de modelos para sistemas dinâmicos segue os seguintes passos: a) projeto do experimento e coleta de dados; b) escolha da estrutura matemática a ser utilizada; c) escolha do critério de desempenho. Com base em (a), (b) e (c), faz-se a estimação de parâmetros do modelo (passo d) via um procedimento de otimização. Neste ponto, passo (d), tem-se o contexto do presente artigo. Por fim, faz-se a validação do modelo estimado (passo e) [1], [4].

Nas últimas décadas, a estimação de parâmetros recebeu uma atenção significativa dentro do procedimento de identificação de sistemas. Dentro do contexto de sistemas lineares, muitas vezes o problema de estimação de parâmetros possui solução fechada. No caso de sistemas não lineares, somente em alguns casos o modelo é linear nos parâmetros o que permite a adoção de procedimentos como o dos mínimos quadrados. Em todos os outros casos, lança-se mão de algoritmos de otimização não linear para calcular o conjunto ótimo de parâmetros, dentro do critério de desempenho selecionado. Embora existam vários tipos de algoritmos para solucionar esse problema, muitos possuem convergência lenta para os ótimos globais quando não convergem para ótimos locais [28],[29],[30].

III. MÉTODOS

A. Evolução Diferencial

Em (Guimarães, 2009; Kushida, Hara, and Takahama, 2015), descrevem a Evolução Diferencial (ED) como uma pesquisa estocástica baseada na população para resolver problemas de otimização no tempo contínuo. A ED é iniciada com a definição de uma população inicial ($X_{j,i,G}$), dentro dos limites superior e inferior para cada parâmetro, esses valores D-dimensional são colocados em dois vetores de inicialização, Bl e Bu, nos quais os índices l e u indicam os limites inferior e superior. A equação (1) mostra o cálculo de $X_{j,i,G}$.

$$X_{j,i,G} = \mathrm{rand}_j(0,1)\left(B_{j,U} - B_{j,L}\right) + B_{j,L}. \quad (1)$$

Com isto, a população passa pela função objetivo onde será minimizada. Este tipo de função é chamada de função de custo e é descrita pela seguinte equação:

$$\text{Min. } f\left(X_{i,G}\right) = f\left(x_1, x_2, \ldots, x_{N_p}\right). \quad (2)$$

Após avaliar cada indivíduo, a população passa pelas seguintes etapas: mutação, cruzamento e seleção. A mutação (F) tem sua faixa de operação definida por $F \in [0{,}5; 1]$. Esta operação possui várias estratégias para realizar a mutação, a seguir são descritas alguns tipos de estratégias [6], [33], [34].

De/rand/1: $V_{i,G} = \alpha_{i,j} + F\left(\beta_{i,j} - \gamma_{i,j}\right)$, (3)

De/rand/2: $V_{i,G} = \alpha_{i,j} + F\left(\beta_{i,j} - \gamma_{i,j}\right) + F\left(\delta_{i,j} - \varepsilon_{i,j}\right)$, (4)

De/best/1: $V_{i,G} = X_{best} + F\left(\alpha_{i,j} - \beta_{i,j}\right)$, (5)

De/current to best/1: $V_{i,G} = X_{i,j} + F.\left(X_{best} - X_{i,j}\right) + F.\left(\alpha_{i,j} - \beta_{i,j}\right)$. (6)

Sendo $\alpha, \beta, \gamma, \delta$ e ε são valores inteiros aleatórios gerados por randi, $i = 1,2, \ldots N_p$, j é o tamanho da população, X_{best} é o valor de aptidão do melhor indivíduo encontrado e F é a taxa de mutação que será aplicada na população [7], [18], [35].

A etapa cruzamento ou recombinação discreta constrói vetores de teste de valores de parâmetros que foram copiados de dois vetores diferença. Sua probabilidade, $Cr \in [0{,}7; 1]$, é um valor definido pelo usuário que controla os valores de parâmetros copiados do mutante. A seguir é descrito o critério do cruzamento:

$$\begin{cases} V_{i,j,G}, & \text{rand}(0,1) \leq Cr \text{ ou } j = j_{rand} \\ X_{i,j,G}, & \text{caso contrário} \end{cases}. \tag{7}$$

onde $V_{i,j,G}$ é a população que sofreu mutação e avaliada pela função objetivo , Cr é a taxa de cruzamento, j_{rand} é um valor gerado aleatoriamente no intervalo [0;1] e $X_{i,j,G}$ é a população atualizada e avaliada pela função objetivo. Por fim, a última etapa desta metaheurística é a seleção, determinando o indivíduo da população como apto ou não através do método de Roleta [36]. Este tipo de método é gerado por uma jogada de roleta, que geralmente processa a quantidade de vezes da população inicial declarada e retorna o índice do melhor indivíduo que tem maiores chances de ser selecionado [37], [38]. A F_T é a soma da função aptidão dos indivíduos que é gerada pela equação (8), (9) e (10).

$$F_T = \sum_{i=1}^{Nind} f_i\,(x). \tag{8}$$

A partir disto, p_i é a probabilidade de seleção de cada indivíduo ser selecionado pela jogada da roleta, sendo descrita pela equação (9).

$$p_i(x) = \frac{f_i(x)}{F_T}\,. \tag{9}$$

Com isto, a probabilidade de acumulação de cada indivíduo com classificação de pior aptidão em relação ao indivíduo de melhor aptidão é descrita pela equação (10).

$$c_i = \sum_{k=1}^{i} p_k\,, \quad i = 1, \ldots, N_{ind}, \tag{10}$$

A acumulação é a somatória da aptidão dos indivíduos e N_{ind} é obtido por um número aleatório no intervalo [0;1].

B. Algoritmo de Otimização de Leão

O Algoritmo de Otimização de Leão (AOL) é um algoritmo desenvolvido recentemente e que imita a vida social dos leões, descrito pelas seguintes etapas: população; mutação; cruzamento; e disputa de território. Leões são os felinos selvagens que mais exibem seu nível de cooperação e força, divididos em dois grupos: leões residentes e leões nômades. O residente é aquele que governa o grupo composto por leoas e filhotes. Mas quando os filhotes atingem à idade adulta, são excluídos do grupo e dando início ao grupo nômade. Eles podem voltar para disputar o território com o leão rei do grupo que foram expulsos. Caso o nômade ganhe a batalha, o rei é expulso ou morto pelo nômade e vice-versa [14], [24], [39].

A população é gerada de forma aleatória e apresentada pela equação matemática (1). Após isto, a população é dividida entre leoas e leões com as seguintes porcentagens: leoas 70% a 90% da população inicial e o restante, 30% a 10%, são os leões.

$$leoa_{i,j} = X_{j,i,G} \leq r{=}85\%, \tag{11}$$

$$leão_{i,j} = X_{j,i,G} > r{=}15\%. \tag{12}$$

Onde $X_{j,i,G}$ representa a população corrente gerada aleatoriamente e r é um vetor no intervalo [0;100] que gera um número aleatório para diferenciar leões e leoas. Após a população ser gerada, ocorre o processo de avaliação da aptidão dos indivíduos para poder selecionar os melhores indivíduos levando ao processo de cruzamento e consequente mutação. O processo de cruzamento (Cr) é representado a partir das seguintes equações:

$$Filho_1 = (Cr)leoa_j + \frac{(1-Cr)}{\sum_{i-1}^{NR} S_i} * leão_{R,i,j} \times S_i, \tag{13}$$

$$Filho_2 = (1 - Cr)leoa_j + \frac{(Cr)}{\sum_{i-1}^{NR} S_i} * leão_{R,i,j} \times S_i. \tag{14}$$

Cada $leoa_j$ é escolhida de acordo com sua aptidão e pode ter de dois a quatro filhotes do mesmo leão, j é a dimensão do problema, i é o tamanho da população, Cr é a taxa de cruzamento dentro do intervalo [0,6;1], NR é o número de leões residentes, $leão_{R,i,j}$ são os leões residentes que predominam no grupo como os mais fortes e S_i é o número de pais selecionados pelo método de seleção. Com isto, a população é atualizada e passa pelo processo de mutação para prevenir a convergência dos ótimos locais. Este processo é realizado de forma semelhante ao do algoritmo genético e a partir da taxa de mutação ocorrerão as alterações nos genes dos indivíduos [40].

$$\text{Mutação} = \{f(\text{Pop}_{\text{cruzada}}),\ j_{\text{rand}} < F. \tag{15}$$

Onde $f(\text{Pop}_{\text{cruzada}})$ é a aptidão dos filhos gerados, j_{rand} é um número aleatório no intervalo [0 1] e F é a taxa de mutação que está no intervalo [0,1;0,4]. Dessa maneira, a população atual é atualizada. Quando os filhotes atingem a idade adulta, considerada quatro anos, eles são expulsos do grupo formando um novo grupo chamado Nomade_{ij}.

$$\text{Leão}_{\text{nomade}} = \text{Nomade}_{ij},\quad \text{se rand}_j > \text{pr}_i, \tag{16}$$

$$\text{Leão}_{\text{nomade}} = \text{Nomade}_{ij},\quad \text{senão rand}_j. \tag{17}$$

Sendo j é a dimensão do problema, i o tamanho da população e rand_j o número aleatório gerado. Com isso, pr_i que é a probabilidade de cálculo de cada $\text{Leão}_{\text{nomade}}$.

$$\text{pr} = 0{,}1 + \min\left(0{,}5\,\frac{(\text{Nomade}_i - \text{Best}_{\text{nomade}})}{\text{Best}_{\text{nomade}}}\right). \tag{18}$$

O $\text{Best}_{\text{nomade}}$ é o leão nômade de melhor aptidão da população Nomade_i. Este passo atualiza a população podendo ou não substituir o leão residente se for pior que ele.

C. Busca Local Iterativa

A Busca Local Iterativa (BLI) realiza a busca por ótimos locais que muitas vezes não são encontradas nas metaheurísticas clássicas. A vantagem deste algoritmo é buscar uma solução ao redor da vizinhança onde tem a chance de obter um ótimo local que muitas vezes outros algoritmos, por serem tão gananciosos ao encontrar soluções de ótimos globais, acabam desprezando uma boa solução encontrada [41].

A BLI inicializa-se pela população através da equação (1) apresentada anteriormente. Com isto é realizada a avaliação dos indivíduos pela função objetivo para poder realizar a busca por ótimos locais através da distância determinada pelo tamanho da população. A busca consegue alcançar os ótimos locais não encontrados até o exato momento. O algoritmo 1 sintetiza a metaheurística descrita a cima.

Pseudocódigo 1: Busca Local Iterativa

$o_0 = \text{População inicial}$
$o^* = \text{Busca Local}\ (o_0)$
Repetir;
$o' = \text{Pertubação}(o^*, \text{hist})$
$o^{*'} = \text{Busca Local}(o')$
$o^* = \text{StopCriterio}(o^*, o^{*'}, \text{hist})$

D. Computação Paralela

Os problemas de otimização requerem uma necessidade de recursos mais amplos devido ao tempo de processamento da CPU e consumo da memória. Mesmo que as metaheurísticas sejam eficientes por reduzirem significativamente a complexidade computacional do processo, o objetivo da computação paralela nas metaheurísticas é acelerar o desempenho reduzindo o tempo de processamento [38], [39].

O paralelismo nas metaheurísticas só é viável quando não ocorrem atrasos significativos no tempo de sincronização das etapas da metaheurísticas, o paralelismo acontece nas iterações no loop interno das metaheurísticas e quando ocorre o atraso, o paralelismo perde sua eficiência retardando o desempenho do algoritmo [9]. Geralmente, o paralelismo é aplicado na função objetivo.

Algoritmo 7: Paralelismo

Gerar população aleatoriamente
para ***i*** ←1 até iterações faça
Avaliar aptidão de cada indivíduo da população
fim ***i***

IV. ALGORITMO HÍBRIDO

Os algoritmos híbridos ganharam grande espaço na engenharia devido ao seu potencial de obter resultados de grande relevância. Esses algoritmos são a junção de dois algoritmos de otimização, ou seja, junta as caracte-

rísticas boas de cada algoritmo em um único algoritmo. Com isso, foram desenvolvidos dois algoritmos híbridos para este trabalho. O primeiro híbrido é a junção da Evolução Diferencial com o Algoritmo de Otimização de Leão e o outro algoritmo híbrido é a junção do Algoritmo de Otimização de Leão com a Busca Local Iterativa (BLI). Estes algoritmos fazem trocas de informações, pois suas técnicas para solucionar o problema não são semelhantes, de tal maneira que os deixa mais robustos. Eles são processados de forma paralela acelerando a busca por melhores soluções sem perder sua generalidade. O algoritmo híbrido é apresentado pela Figura 1.

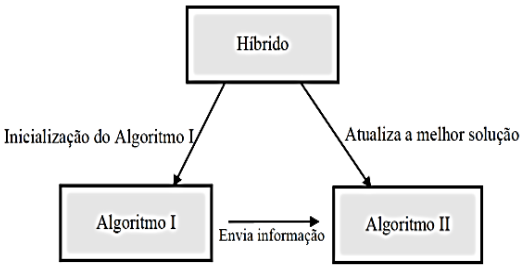

Figura 1. Diagrama do algoritmo híbrido EDP-AOLP.

Estes algoritmos enviam $best_{par}$ um ao outro armazenando as melhores soluções a fim de melhorar a qualidade de pesquisa das metaheurísticas. O Algoritmo I envia $best_{par}$ para o Algoritmo II onde é acrescentado à população inicial do Algoritmo II. Assim, ocorre a troca de informação entre eles.

V. RESULTADOS

Nesta seção, são apresentados os resultados dos métodos descritos neste trabalho que soluciona o problema de estimação de parâmetros em modelos dinâmicos não lineares. Dois estudos de caso são apresentados. O primeiro trata da estimação de parâmetros do modelo de uma turbina do tipo Francis utilizada na geração de energia elétrica em usinas hidrelétricas. Os dados de entrada e saída aqui utilizados foram medidos em campo em uma usina do sistema interligado nacional (SIN). O segundo estudo de caso está baseado na função de teste Rastrigin, a qual é considerada uma função de otimização que analisa o desempenho dos algoritmos ao convergirem para os ótimos dentro do seu espaço de busca, determinado pelo intervalo [-5,12;5,12].

Para fins de comparação do tempo computacional, todos os experimentos foram executados em um computador pessoal com processador Intel i5 – 3570, 3.40 GHz, 8GB de RAM, 930 GB, sistema operacional 64 bits e o software Matlab®.

i) Modelagem do Conjunto Conduto/Turbina de uma usina Hidrelétrica.

Figura 2. Turbina Francis.

Este estudo de caso trata da obtenção de um modelo tipo caixa cinza da potência gerada por uma turbina do tipo Francis em função da abertura do distribuidor que regula a quantidade de água que passa pelo conduto (Turbine Gate). Detalhes sobre este problema são ilustrados na Figura 3.

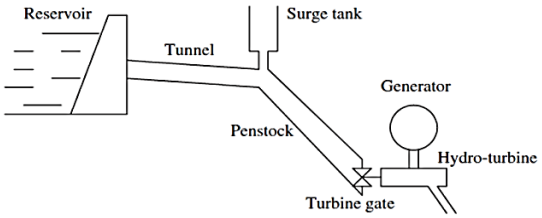

Figura 3. Diagrama do sistema conduto/turbina.
Fonte: Kishor,Saini,Singh, 2007,p 778.

A turbina aqui tratada pode ser visualizada na Figura 3 e pertencente ao sistema interligado nacional (SIN), cujos dados foram obtidos experimentalmente.

O sinal de saída desse sistema é a potência mecânica (P_m) da turbina em pu, o sinal de entrada é a abertura do distribuidor (D_{ist}) em pu, G é a posição do *gate* do distribuidor em pu, q é a vazão em pu, h é a altura em pu, ho é o valor de estado estacionário inicial de h em pu, T_w é a constante de tempo de inércia da água em segundos onde não se tem seu valor, qNL é a vazão necessária para suprir as perdas em vazio da turbina em pu, $1/s$ é o integrador; A_{tpre}, A_t são os respectivos ganhos do sistema onde não se tem seus valores.

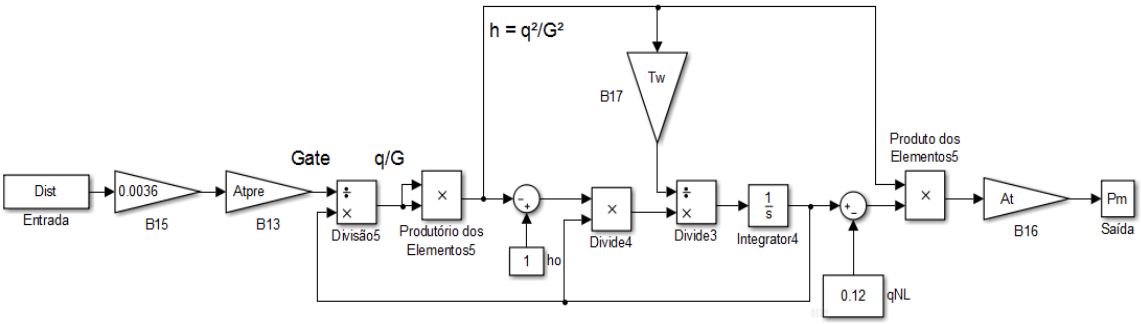

Figura 4. Modelo matemático da Turbina Francis.

A partir do modelo representado na Figura 4 foi possível formular as equações dos três parâmetros (A_{tpre}, T_w e A_t) que são estimados. Mais detalhes sobre modelos caixa cinza deste tipo de sistema dinâmico podem ser encontrados em [44]. Os respectivos espaços de busca destas variáveis são.

$$A_{tpre} = b_L + \text{rand}(np, 1)(b_U - b_L), \quad (18)$$

$$T_W = b_L + \text{rand}(np, 1)(b_U - b_L), \quad (19)$$

$$A_t = b_L + \text{rand}(np, 1)(b_U - b_L). \quad (20)$$

Onde b_U é o limite superior e b_L o limite inferior.

Neste estudo de caso todas as metaheurísticas possuem tamanho da população igual 30 ($N_p = 30$), a dimensão é 3 ($\dim = 3$) e o número máximo de conjuntos de avaliações é 500 (iter = 500). Para determinar as taxas de cruzamento e seleção dos algoritmos foi necessário testá-los aleatoriamente dentro do intervalo ($F = [0{,}5; 1] Cr = [0{,}7; 1]$ para ED e $F = [0{,}1; 1]$ e $Cr = [0{,}5; 1]$ para AOL). Os principais parâmetros das metaheurísticas estão descritos na Tabela 1.

Tabela 1. Parâmetros específicos para as metaheurísticas.

Parâmetro	**ED**	**AOL**
Taxa de mutação	0,5	0,3
Taxa de cruzamento	0,7	0,6
Idade adulta	-	4
Seleção	Roleta	Roleta
Estratégia	$V_{i,G} = \alpha_{i,j} + F.(\beta_{i,j} - \gamma_{i,j})$	Binário

Para o Algoritmo Híbrido I e II, a única alteração realizada está na população inicial ($X_{j,i,G}$), a primeira coluna de $X_{j,i,G}$ é substituída pela melhor solução ($best_{par}$) encontrada anteriormente, passando para o próximo algoritmo. Para uma análise comparativa das metaheurísticas é utilizado o parâmetro estatístico erro médio quadrático (*mean square error*), dado na equação 21.

$$MSE = \sum_{i=1}^{iter}\left(\left|a_i^{Ref} - a_i^{est}\right|\right)^2. \quad (21)$$

Onde a_i^{Ref} é a resposta de referência, a_i^{est} é a resposta obtida pelas metaheurísticas, $iter$ é o número máximo de iterações. Para obter os resultados foram executadas duzentas e cinquenta vezes cada algoritmo e, com isto, foi possível obter a média, o máximo e o mínimo dos parâmetros estimados.

Tabela 2. Parâmetros estimados do conjunto/turbina.

Algoritmo	Parâm	Mín	Máx	Méd	θ
ED	A_{tpre}	0,673	0,998	0,757	0,137
	T_w	1,508	1,992	1,833	0,139
	A_t	1,310	1,913	1,546	0,198
AOL	A_{tpre}	0,630	0,997	0,768	0,012
	T_w	1,423	1,969	1,842	0,136
	A_t	1,351	1,650	1,466	0,043
BLI	A_{tpre}	0,7394	0,997	0,801	0,053
	T_w	0,954	1,994	1,716	0,144
	A_t	0,997	1,672	1,536	0,064
Híbrido I (ED com AOL)	A_{tpre}	0,662	0,999	0,766	0,023
	T_w	1,502	1,994	1,824	0,135
	A_t	1,321	1,849	1,531	0,131
Híbrido II (AOL com BLI)	A_{tpre}	0,619	0,999	0,752	0,100
	T_w	1,502	1,999	1,852	0,111
	A_t	1,355	1,956	1,588	0,036

A Figura 5 (a) e (b) ilustram a saída obtida através dos valores médios de cada parâmetro comparados com a saída dos dados de campo.

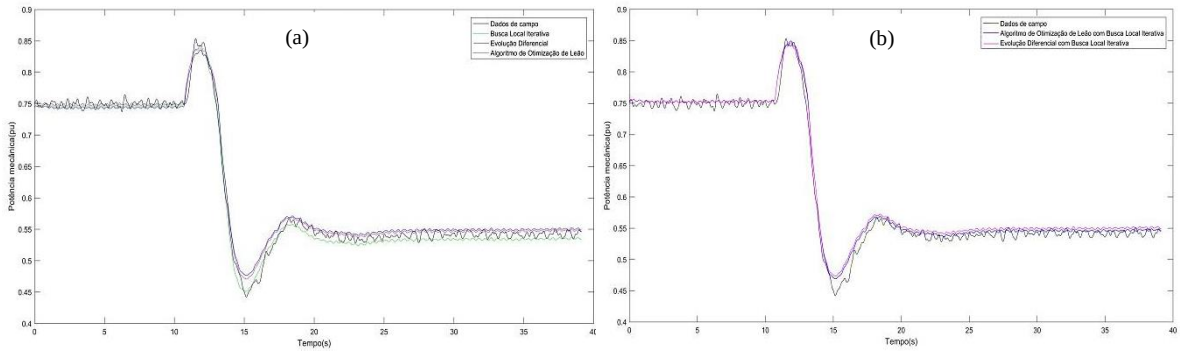

Figura 5. (a) Sinal de saída das metaheurísticas individuais; (b) Sinal de saída das metaheurísticas colaborativas.

A Tabela 3 mostra os valores máximos, mínimos, médios e o desvio padrão (θ) de cada metaheurística testada através da comparação do MSE.

Tabela 3. Desvio padrão de cada metaheurística.

Algoritmo	Mín.	Máx.	Méd.	θ
ED	5,728	47,913	15,523	8,202
AOL	21,765	160,05	76,736	32,718
BLI	27,615	151,16	53,751	18,721
Híbrido I (ED com AOL)	5,703	159,97	18,22	11,8
Híbrido II (AOL com BLI)	5,509	37,741	12,812	5,592

A Figura 6 mostra o MSE, utilizando a ferramenta boxplot.

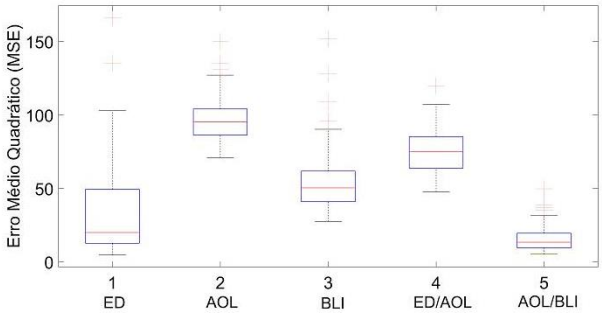

Figura 6. Erro médio quadrático dos modelos calculados por cada método.

Através da Figura 7 (a), (b) e (c) pode se visualizar a variação dos parâmetros estimados por cada metaheurísticas individualmente e com a Figura 7 (d) e (e) estão relacionadas às metaheurísticas Híbrida I e II.

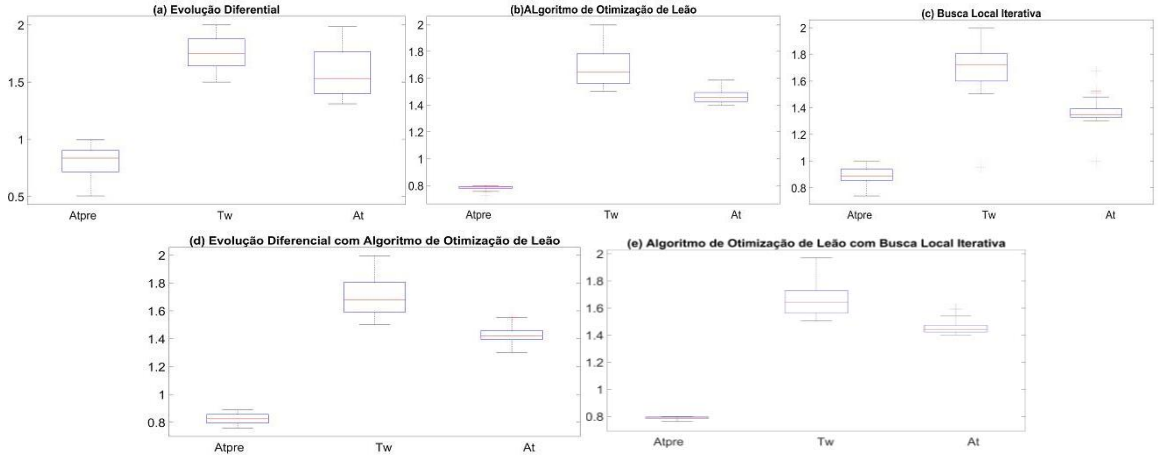

Figura 7. (a) Metaheurística ED; (b) Metaheurística AOL; (c) Metaheurística BLI; (d) Metaheurística ED/AOL; (e) Metaheurística AOL/BLI.

Observando a Figura 7, infere-se que os algoritmos híbridos possuem os valores dos parâmetros próximos dos valores médios, garantindo uma repetibilidade dos valores dos parâmetros.

Por fim, no que diz respeito ao tempo de processamento de cada algoritmo, este é apresentado pela Tabela 4, tomando-se como base 250 execuções com 500 iterações. O tempo obtido de cada metaheurística executada de forma paralela foi medido através do comando *tic-toc* da plataforma Matlab®.

Tabela 4. Tempo de processamento.

Algoritmo	**Tempo**
Evolução Diferencial (ED)	12 min. e 19 seg.
Algoritmo de Otimização de Leão (AOL)	6 min. e 21 seg.
Busca Local Iterativa (BLI)	20 min. e 36 seg.
Híbrido I (ED com AOL)	18 min.
Híbrido II (AOL com BLI)	22 min. e 43 seg.

Nesta comparação os algoritmos realizaram 500 iterações em cada execução, para uma análise mais justa. O Algoritmo de Otimização de Leão teve o melhor desempenho em relação ao tempo computacional gasto.

ii) Função de teste Rastrigin

Na literatura existem diversos tipos de funções de otimização para testar a eficiência de um algoritmo, neste estudo de caso foi escolhida a função Rastrigin para tratar de dimensionalidades grandes.

$$f(x) = 10d + \sum_{i=1}^{d}[x_i^2 - 10\cos(2\pi x_i)]. \quad (21)$$

Sendo d o número de iterações máxima, $i = 1...d$, x_i é selecionado aleatoriamente dentre os indivíduos da população. Para obter os resultados este estudo de caso utilizou o tamanho da população igual 30 ($np = 30$), a dimensão 10 ($dim = 10$), o número máximo de conjuntos de avaliações 500 ($iter$ = 500) e duzentas e cinquenta execuções.

Tabela 5. Parâmetros obtidos dentro do espaço de busca.

Algoritmo	Mín	Máx	Méd	θ
ED	-5,112	5,075	0,009	0,1418
AOL	-5,116	5,062	-0,013	0.1533
BLI	-5,095	5,102	0,020	0,0770
Híbrido I (ED com AOL)	-5,075	5,046	0,0086	0,1509
Híbrido II (AOL com BLI)	-5,093	5,0557	-0,0297	0,1720

Conforme os dados demonstrados pela Tabela 5, pode se observar a robustez das metaheurísticas ao encontrar as melhores soluções ótimas dentro do espaço de busca. Para uma comparação entre estas metaheurísticas, a seguir é demonstrado pela Figura 8 a variação das melhores soluções obtidas durante toda execução.

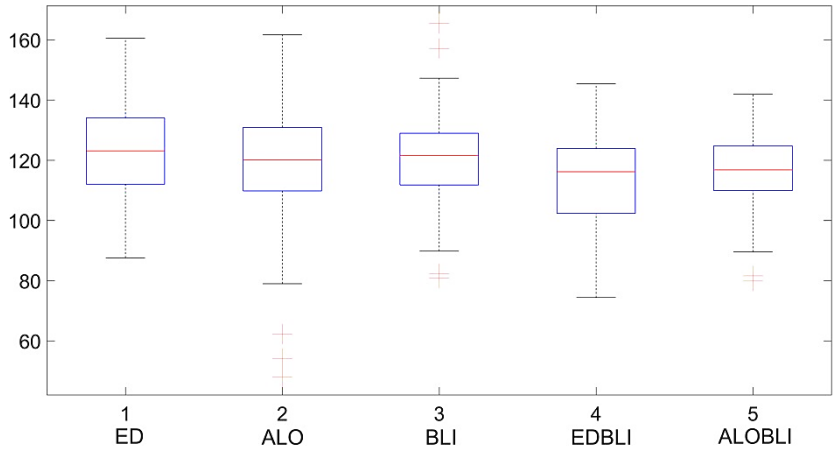

Figura 8. Erro médio quadrático da função de teste Rastrigin.

A variação média destas técnicas de otimização está no centro da caixa referenciada pelo traço vermelho e as extremidades altas e baixas referência os valores mínimos e máximos alcançados. Ao visualizar a Figura 9, os pontos médios de todas as metaheurísticas estão próximos uns dos outros, que estão entre 110 a 125. Em relação aos híbridos I e II são bons estimadores de parâmetros conforme os dados coletados da Tabela 5.

Tabela 6. Variância do MSE.

Algoritmos	Min	Máx	Méd	θ
Evolução Diferencial (ED)	87,557	160,447	122,524	14,726
Algoritmo de Otimização de Leão (AOL)	47,925	161,638	118,793	19,476
Busca Local Iterativa (BLI)	80,903	165,442	121,030	14,73
Híbrido I (ED com AOL)	63,805	148,681	110,004	19,832
Híbrido II (ED com BLI)	79,852	141,995	116,417	12,005

Em relação ao tempo computacional gasto a Tabela 7 demonstra os respectivos tempos de cada metaheurística.

Tabela 7. Tempo computacional gasto

Algoritmo	Tempo
Evolução Diferencial (ED)	9 min. e 14 seg.
Algoritmo de Otimização de Leão (AOL)	4 min. e 58 seg.
Busca Local Iterativa (BLI)	15 min.
Híbrido I (ED com AOL)	23 min. e 10 seg.
Híbrido II (AOL com BLI)	20 min. e 59 seg.

A partir das Tabelas 7 e 4, o tempo computacional depende da quantidade de dados a serem utilizados e a quantidade de parâmetros a serem estimados em ambos os estudos de caso. No segundo estudo de caso, o Híbrido I teve um aumento de aproximadamente 1,2% em relação ao Híbrido I utilizado no primeiro estudo de caso.

VI. CONCLUSÃO

Este trabalho teve como premissa a aplicação de metaheurísticas na estimação de parâmetros visando melhorar e diminuir o tempo computacional através dos dois algoritmos híbridos propostos.

O algoritmo híbrido II proposto neste trabalho apresentou soluções que possuíam uma menor variação dos parâmetros, ou seja, uma maior repetibilidade dos valores estimados, mostrando ser um algoritmo robusto para a estimação de parâmetros. Através do estudo de caso da função Rastrigin ratifica-se que o algoritmo proposto tem um desempenho superior aos demais. Desta forma, o algoritmo proposto é uma alternativa para o problema de estimação de parâmetros em sistemas não lineares.

Embora o Híbrido I apresentou um aumento computacional em relação ao primeiro estudo de caso, este não deixou de encontrar boas soluções.

REFERÊNCIAS

[1] L. A. Aguirre, "Introdução à Identificação de Sistemas – Técnicas Lineares e Não-Lineares," in *UFMG*, 4ª edição., UFMG, Ed. Belo Horizonte, 2015, p. 730.

[2] L. Ljung, *System Identification Theory for User*. New Jersey, 1987.

[3] A. A. R. Coelho and L. dos S. Coelho, *Identificação de Sistemas Dinâmicos Lineares*, 1ª. Florianópolis, SC, 2004.

[4] M. V. Corrêa and A. L., "Identificação Não-linear Caixa-cinza: Uma Revisão e Novos Resultados," *Rev. Control. Automação*, vol. 15, no. 2, p. 18, 2004.

[5] L. M. Margoti, A. P. L. Santos, N. R. L. Milagres, A. A. Campos, G. F. V. Amaral, and M. F. S. Barroso, "Aplicação de Representações em Blocos Interconectados em Identificação Caixa-cinza de Sistemas Dinâmicos Não-lineares," pp. 4224–4230, 2015.

[6] R. Storn and K. Price, "Differential Evolution -- A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *J. Glob. Optim.*, vol. 11, pp. 341–359, 1997.

[7] X. Zhou, G. Zhang, X. Hao, L. Yu, and D. Xu, "Differential Evolution with Multi-Stage Strategies for Global Optimization," pp. 2550–2557, 2016.

[8] S. Mirjalili, "The Ant Lion Optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, 2015.

[9] T. Bäck, D. B. Fogel, and Z. Michalewicz, "Handbook of Evolutionary Computation," *Evol. Comput.*, vol. 2, pp. 1–11, 1997.

[10] L. dos S. Coelho, "Notas em Matemática Aplicada," *Soc. Bras. Matemática Apl. e Comput.*, vol. 2, p. 99, 2003.

[11] E. Paterakis, V. Petridis, and A. Kehagias, "Genetic Algorithm in Parameter Estimation of Nonlinear Dynamic Systems," *{P}arallel {P}roblem {S}olving from {N}ature -- {PPSN V}*, pp. 1008–1017, 1998.

[12] X. Yan, "A hybrid algorithm based on particle swarm optimization and group search optimization," pp. 13–17, 2011.

[13] B. Dong, A. Zhou, and G. Zhang, "A Hybrid Estimation of Distribution Algorithm with Differential Evolution for Global Optimization," no. 1, 2016.

[14] S. Mirjalili, "The Ant Lion Optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, 2015.

[15] R. Satheeshkumar and R. Shivakumar, "Ant Lion Optimization Approach for Load Frequency Control of Multi-Area Interconnected Power Systems," no. July, pp. 2357–2383, 2016.

[16] A. E. Eiben and M. Schoenauer, "Evolutionary computing," *Inf. Process. Lett.*, vol. 82, no. 1, pp. 1–6, 2002.

[17] H. De Garis, *Introduction to Evolutionary Computing*, vol. 12. 2004.

[18] S. R. M. Lampinen J. A., Price K. V., *Differential Evolution - A Practical Approach to Global Optimization*. New York, NY, USA, 2005.

[19] M. Yang, Z. Cai, C. Li, and J. Guan, "An improved JADE algorithm for global optimization," *Proc. 2014 IEEE Congr. Evol. Comput. CEC 2014*, pp. 806–812, 2014.

[20] M. Clerc, *Particle Swarm Optimization*. John Wiley & Sons, 2010.

[21] R. K. Mallick, "Hybrid Differential Evolution Particle Swarm Optimization (DE-PSO) algorithm for optimization of Unified Power flow controller parameters," pp. 635–640, 2016.

[22] H. R. Lourenço, O. C. Martin, and T. Stützle, "Chapter 11 ITERATED LOCAL SEARCH," *ReCALL*.

[23] D. Dibblee, J. Maltese, B. M. Ombuki-Berman, and A. P. Engelbrecht, "Vector-Evaluated Particle Swarm Optimization With Local Search," *2015 IEEE Congr. Evol. Comput.*, pp. 187–195, 2015.

[24] B. R. Rajakumar, "The Lion's Algorithm: A New Nature-Inspired Search Algorithm," *Procedia*

Technol., vol. 6, pp. 126–135, 2012.
[25] M. Yazdani and F. Jolai, "Lion Optimization Algorithm (LOA): A nature-inspired metaheuristic algorithm," *J. Comput. Des. Eng.*, vol. 3, no. 1, pp. 24–36, 2016.
[26] C. Blum, "Hybrid Metaheuristics in Combinatorial Optimization: A survey," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7505 LNCS, pp. 1–10, 2012.
[27] L. A. Aguirre, A. H. Bruciapaglia, P. E. Miyagi, and J. R. C. Piqueira, *Enciclopédia de automática : controle e automação*. Blucher, 2007.
[28] J. Chen, Y. Zhang, and R. Ding, "Auxiliary model based multi-innovation algorithms for multivariable nonlinear systems," *Math. Comput. Model.*, vol. 52, no. 9–10, pp. 1428–1434, 2010.
[29] C. Wang and L. Zhu, "Parameter identification of a class of nonlinear systems based on the multi-innovation identification theory," *J. Franklin Inst.*, vol. 352, no. 10, pp. 4624–4637, 2015.
[30] J. Li and F. Ding, "Parameter fitting for nonlinear systems," 2011.
[31] F. G. Guimarães, "Algoritmos de evolução diferencial para otimização e aprendizado de máquina," *An. do IX Congr. Bras. Redes Neurais*, pp. 1–17, 2009.
[32] J. I. Kushida, A. Hara, and T. Takahama, "Rank-based differential evolution with multiple mutation strategies for large scale global optimization," *2015 IEEE Congr. Evol. Comput. CEC 2015 - Proc.*, no. 1, pp. 353–360, 2015.
[33] Y. Ao and H. Chi, "Experimental Study on Differential Evolution Strategies," *2009 WRI Glob. Congr. Intell. Syst.*, pp. 19–24, 2009.
[34] B. Yuan, B. Li, H. Chen, and X. Yao, "A new evolutionary algorithm with structure mutation for the maximum balanced biclique problem," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1040–1053, 2015.
[35] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, 2009.
[36] S. M. Bruschi and R. F. D. Mello, "Algoritmo genético para a tomada de decisão de particionamento de processos lógicos em simulação distribuída," *IEEE Lat. Am. Trans.*, vol. 6, no. 1, pp. 97–105, 2008.
[37] R. Das, "Parameter Estimation of a Space Radiator using Differential Evolution Algorithm," 2016.
[38] A. Lipowski and D. Lipowska, "Roulette-wheel selection via stochastic acceptance," *Phys. A Stat. Mech. its Appl.*, vol. 391, no. 6, pp. 2193–2196, 2012.
[39] B. Wang, X. P. Jin, and B. Cheng, "Lion pride optimizer: An Optimization Algorithm Inspired by Lion Pride Behavior," *Sci. China Inf. Sci.*, vol. 55, no. 10, pp. 2369–2389, 2012.
[40] B. R. Rajakumar, "The Lion's Algorithm: A New Nature-Inspired Search Algorithm," *Procedia Technol.*, vol. 6, pp. 126–135, 2012.
[41] W. A. X. Melo, M. H. C. Fampa, and F. M. P. Raupp, "BUSCA LOCAL INTENSIVA: UMA NOVA METAHEURÍSTICA PARA OTIMIZAÇÃO GLOBAL RESTRITA," *An. do XLI Simpósio Bras. Pesqui. Operacional, SBPO*, 2009.
[42] S. M. Sait, K. S. Khan, and M. I. Ali, "Parallel strategies for stochastic evolution," *Proc. 7th Int. Conf. Intell. Syst. Des. Appl. ISDA 2007*, pp. 813–818, 2007.
[43] E. G. Talbi, *Metaheuristics: From Design to Implementation*. 2009.
[44] N. Kishor, R. P. Saini, and S. P. Singh, "A review on hydropower plant models and control," *Renew. Sustain. Energy Rev.*, vol. 11, no. 5, pp. 776–796, 2007.