

EVOLUÇÃO DIFERENCIAL OTIMIZADO POR LÓGICA NEBULOSA

Allan C. Krainski Ferrari, Gideon Villar Leandro, Gustavo H. C. Oliveira, Eduardo Parente Ribeiro
Programa de Pós-Graduação em Engenharia Elétrica (PPGEE)
Universidade Federal do Paraná (UFPR)
Curitiba, Brasil

Abstract—This work applies changes in the differential evolution algorithm via fuzzy logic, in order to reduce the computational effort in the identification process of a multivariable system. This work is carried out in two stages. In the first stage, the differential evolution algorithm is simulated several times to obtain the best values: cross rate, mutation rate and population size. This step is also determined the best model of representation of multivariable system that is used in the second stage. In the second stage, the modifications based on fuzzy logic are applied. These modifications adjust the values of cross rate and mutation rate, in real time. The results show that the modifications implemented improve the efficiency of the conventional differential evolution algorithm.

Keywords—*differential evolution; multivariable system; fuzzy logic; turning parameters online*

I. INTRODUÇÃO

Todos dos sistemas físicos reais do mundo são multivariáveis, onde o comportamento desses sistemas depende da interação simultânea de várias variáveis. Muitas vezes, os sistemas multivariáveis são modelados e representados por um modelo SISO (*simple input and simple output*), sendo que o modelo tem uma variável de entrada e outra de saída. A influência de outras variáveis na dinâmica do sistema SISO é fraca, sendo considerada como perturbações. Enquanto em outros casos, essa influência não é fraca e o comportamento de um sistema depende da interação simultânea de várias variáveis. Nesse caso a representação do modelo depende de múltiplas entradas e múltiplas saídas, denominado de sistema MIMO (*multiple inputs and multiple output*) [1]. Além da multivariabilidade que pode estar presente na forma de interação ou acoplamento entre sinais de entrada e saída, um sistema real depende também da ordem da linearidade ou não linearidades presentes na dinâmica do sistema [2].

A identificação de um sistema físico pode ser desafiador, pois consiste em representar seu comportamento por um modelo matemático. Um sistema físico pode ser modelado de forma simplificada por um modelo linear para uma determinada faixa de operação. Quando o modelo linear não é o suficiente para representar um comportamento é necessário utilizar modelos não lineares [2].

Dentre as várias abordagens para a modelagem matemática sistemas, existem os algoritmos de otimização determinísticos e os estocásticos. Os algoritmos determinísticos diferem dos estocásticos por usar a informação do gradiente da função

objetivo, porém quando a função objetivo apresenta descontinuidades, esses algoritmos podem não convergir para os valores esperados, isto é, os ótimos globais. Nesse caso é preferível utilizar algoritmos que não usam a informação do gradiente, mas somente valores de funções como no caso dos algoritmos estocásticos [3]. Dentre os algoritmos estocásticos destacam-se os meta-heurísticos que têm seu processo de otimização inspirado no comportamento de um processo natural, por exemplo: colônia de abelhas, formigas, grupos de morcegos, algoritmos genéticos redes neurais artificiais, etc [4].

Um dos algoritmos baseados em comportamento natural para estimação dos parâmetros de processos físicos é o algoritmo da Evolução Diferencial (ED) [5]. O algoritmo ED é robusto e tem rápida convergência na busca das soluções desejadas. A estratégia de otimização desse algoritmo se baseia na evolução da população de soluções que representam os parâmetros possíveis da função desejada a ser otimizada. A evolução desses parâmetros acontece por meio de processos de cruzamento, mutação e seleção [5].

Na literatura, encontram-se várias aplicações do algoritmo ED. Por exemplo, o algoritmo ED é aplicado na identificação de sistemas multivariáveis [6], [7]. Em outros trabalhos [4], [8], o algoritmo ED tem seus processos de mutação e cruzamento modificados visando melhorar a convergência do algoritmo. Também pode ser encontrado o algoritmo ED implementado a outras estratégias de identificação como de redes neuronais artificiais [9].

Este trabalho visa diminuir o número de interações do algoritmo ED e conseqüentemente seu custo computacional no processo de identificação por meio de modificações no processo de cruzamento e mutação do algoritmo ED por lógica *fuzzy*. As modificações são colocadas à prova na identificação de em um sistema multivariável real não linear.

Este artigo está estruturado em três partes: método, resultados e discussões e conclusão. No método é explicado como as modificações no algoritmo ED por lógica *fuzzy* são aplicadas. Nos resultados e discussões é mostrado o melhor modelo que representa o sistema físico real e o desempenho dos algoritmos propostos, depois é feita uma análise dos mesmos. Finalmente na conclusão são apresentados as restrições e os desdobramentos para trabalhos futuros.

II. MÉTODO

Assim como outros algoritmos evolucionários, o algoritmo ED utiliza uma população de soluções candidatas randomicamente geradas, dentro de uma região de domínio, para minimizar a função objetivo. À medida que a população evolui, a direção de busca e o tamanho do passo na mutação mudam auto ajustando-se de acordo com a distribuição da população no espaço de busca [5].

A evolução diferencial pode ser considerada como um desenvolvimento adicional para algoritmos genéticos. Em relação ao algoritmo genético, a evolução diferencial considera as soluções como série de números reais sem precisar usar processos de codificação ou decodificação [3].

O algoritmo ED possibilita ao usuário a escolha do tipo de estratégia que será utilizada durante o processo evolutivo. Tais estratégias dependem do vetor escolhido para ser perturbado, do número de vetores que serão considerados para a perturbação e do tipo de cruzamento [5].

Também há aplicações de técnicas que ajustem as taxas de cruzamento e mutação do algoritmo ED em tempo real. Na literatura são encontrados trabalhos como o algoritmo JADE [8] e o algoritmo ED por voo de Lévy [4] entre outros que visam diminuir o número de interações na estimação. Neste trabalho, a adaptação no processo de cruzamento e mutação é realizada por lógica *fuzzy*.

A lógica *fuzzy* é uma técnica de inteligência artificial que provê um método de traduzir expressões verbais, vagas, imprecisas e qualitativas, comuns a comunicação humana, em valores numéricos. Sua metodologia e seus conceitos são fáceis de entender e aplicar, mas o usuário necessita ter um bom conhecimento do sistema a ser estudado [10].

No método *fuzzy* para ajustadores de parâmetros utilizado neste trabalho são aplicadas duas topologias: a primeira recebe o valor do desempenho da evolução da população a cada geração, sendo este representado por taxa de melhoria (*IR*); a segunda recebe o valor da relação entre a geração atual e número total de gerações, sendo este representado por momento (*MoM*) [11].

A taxa de melhoria (*IR*) é calculada pelo *fitness* da função objetivo da evolução da população de soluções da geração atual do algoritmo ED.

$$IR = \frac{\text{fitness da geração atual} - \text{fitness da geração anterior}}{\text{fitness da geração anterior}} \quad (1)$$

A função *fitness* indica a qualidade de uma solução estimada em relação ao comportamento de um sistema real. Quanto maior o valor do *fitness*, melhor sua modelagem. O inverso do valor do *MSE* (*Medium of Squared Error*) do melhor indivíduo da população do algoritmo ED é usado como função *fitness*.

O momento (*MoM*) é determinado pela razão do valor da geração atual e pelo número máximo de gerações.

$$MoM = \frac{\text{Geração atual}}{\text{Número máximo de gerações}} \quad (2)$$

O modelo do ajustador *fuzzy* dos parâmetros das taxas de cruzamento e mutação recebe na entrada o valor correspondente a *IR* ou *MoM*. Em sua saída é gerado o valor correspondente à taxa de cruzamento (C_R) e a taxa de mutação (F). Esses valores são fuzzificados por suas respectivas funções de pertinência e são obtidos os seus valores linguísticos.

Com os valores *IR* e *MoM*, determinam-se os valores semânticos de C_R , F e seus respectivos graus de pertinências. Depois de determinado estes valores, é aplicado o processo de defuzzificação para estimar os valores numéricos C_R e F . As modificações *fuzzy* no algoritmo ED estão representadas no fluxograma da Fig. 1.

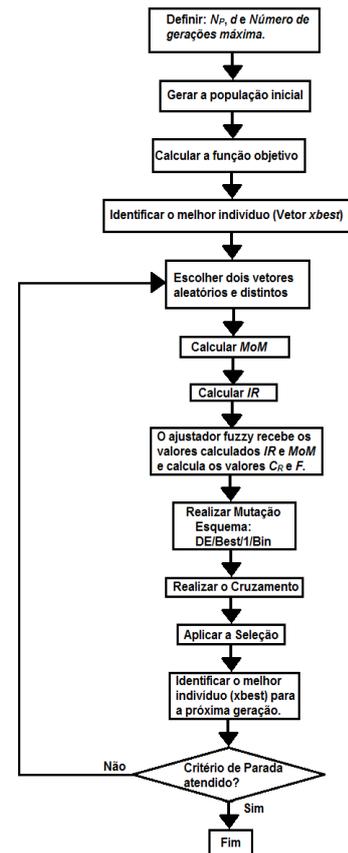


Fig. 1: Fluxograma do algoritmo ED modificado por lógica *fuzzy*.

O sistema *fuzzy* do modelo dos ajustadores das taxas de cruzamento e mutação deste trabalho utiliza o modelo Mamdani [10], funções de pertinência triangular, operador de agregação de antecedentes mínimo, operador de agregação de regras máximo e defuzzificação por centro do máximo.

Os termos linguísticos das funções de pertinências, utilizados para descrever os valores semânticos da variável de entrada *IR* são declarados como: BA (baixo); BM (baixo-

médio); MD (médio); MA (médio-alto) e AL (alto). Para descrever os valores semânticos da variável de entrada *MoM* são declarados como: IN (início); IM (início-meio); ME (meio); MF (meio-fim) e FM (fim). Os valores semânticos para as variáveis de saída C_R e F são declarados como: PQ (pequeno); PM (pequeno-médio); MD (médio); MG (médio-grande) e GD (grande).

Neste trabalho, o processo de mutação do algoritmo ED usa o esquema *ED/Best/1/Bin*, no qual a população evolui baseada na direção do melhor indivíduo. O algoritmo ED foi modificado em três esquemas: o primeiro está em função de *MoM*; o segundo está em função de *IR* e o terceiro aplica os dois otimizadores *fuzzy*. No terceiro esquema geram-se dois vetores mutação com valores de C_R e F distintos, um vetor utiliza esses valores em função de *MoM* e o outro em função de *IR*.

III. RESULTADOS

Para a identificação foi utilizado um sistema físico real multivariável se constituído de uma entrada e duas saídas. A variável de entrada é a frequência do inversor de frequência que controla a bomba que alimentam o reservatório. As variáveis de saída são fornecidas pelos sensores de pressão e vazão que monitoram o comportamento de atuação das bombas. Na Fig. 2 estão os dados utilizados no processo de estimação e validação.

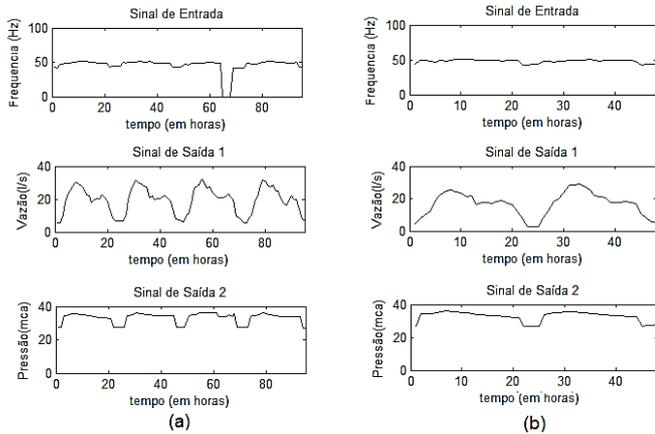


Fig. 2: Dados do sistema SIMO: (a) estimação e (b) validação.

Este sistema é denominado de sistema SIMO (*simple inputs and multiple output*). É considerado desafiador para a identificação, pois seu comportamento depende da interação simultânea de várias variáveis [1].

Os dados foram fornecidos pela companhia paranaense de saneamento (SANEPAR) [12] e tem tempo de amostragem igual à uma hora.

Para o processo de identificação foram usados o modelo linear e os modelos de não linearidade estática que estão representados em espaço de estados em tempo discreto. Os modelos não lineares são: o modelo de Hammerstein, Wiener e Hammerstein-Wiener. Na equação (3), está representado o Hammerstein-Wiener. O modelo de Hammerstein pode ser extraído ignorando-se as não linearidades estáticas da saída, ou seja, fazendo: $Y(k) = W(k)$. Para obter modelo de Wiener,

ignoram-se as não linearidades presentes na entrada do processo, ou seja, fazendo: $X(k) = u(k)$. Estes modelos são utilizados porque conseguem representar de forma eficaz as não linearidades estáticas presente em atuadores [13].

$$X[k] = [x_1 \cdot u[k] + x_2 \cdot u[k]^2 + x_3 \cdot u[k]^3 + x_4 \cdot u[k]^4 + x_5 \cdot u[k]^5]_{1 \times 1}$$

$$\begin{aligned} x[k+1] &= Gx[k] + H \cdot X[k] \\ W[k] &= Cx[k] + DX[k] \end{aligned}$$

$$Y[k] = \begin{bmatrix} y_{11} \cdot w_1[k] + y_{12} \cdot w_1[k]^2 + y_{13} \cdot w_1[k]^3 + y_{14} \cdot w_1[k]^4 + y_{15} \cdot w_1[k]^5 \\ y_{21} \cdot w_1[k] + y_{22} \cdot w_1[k]^2 + y_{23} \cdot w_1[k]^3 + y_{24} \cdot w_1[k]^4 + y_{25} \cdot w_1[k]^5 \end{bmatrix}_{2 \times 1} \quad (3)$$

Onde:

$$x[k+1] = \begin{bmatrix} x_{11}[k+1] \\ x_{21}[k+1] \\ x_{31}[k+1] \\ x_{12}[k+1] \\ x_{22}[k+1] \\ x_{32}[k+1] \end{bmatrix}_{6 \times 1}, \quad x[k] = \begin{bmatrix} x_{11}[k] \\ x_{21}[k] \\ x_{31}[k] \\ x_{12}[k] \\ x_{22}[k] \\ x_{32}[k] \end{bmatrix}_{6 \times 1}$$

$$G = \begin{bmatrix} 0 & 0 & -a_{31} & 0 & 0 & 0 \\ 1 & 0 & -a_{21} & 0 & 0 & 0 \\ 0 & 1 & -a_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -a_{32} \\ 0 & 0 & 0 & 1 & 0 & -a_{22} \\ 0 & 0 & 0 & 0 & 1 & -a_{12} \end{bmatrix}_{6 \times 6}$$

$$Y[k] = \begin{bmatrix} y_1[k] \\ y_2[k] \end{bmatrix}_{2 \times 1}, \quad C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{2 \times 6}$$

$$H = \begin{bmatrix} (b_3 - a_{31}b_0) \\ (b_2 - a_{21}b_0) \\ (b_1 - a_{11}b_0) \\ (b_3 - a_{32}b_0) \\ (b_2 - a_{22}b_0) \\ (b_1 - a_{12}b_0) \end{bmatrix}, \quad D = \begin{bmatrix} b_{01} \\ b_{01} \end{bmatrix}_{2 \times 1} \quad e \quad W[k] = \begin{bmatrix} w_1[k] \\ w_2[k] \end{bmatrix}_{2 \times 1}$$

Inicialmente, foi realizado um conjunto de testes com o objetivo de obter os melhores valores de C_R , F , o tamanho da população do algoritmo ED e o modelo que representa o sistema SIMO. Os intervalos utilizados para os parâmetros de cruzamento e mutação foram $C_R \in [0 \ 1]$ e $F \in [0 \ 1]$, a população N_p começou com 100 indivíduos, depois este processo foi repetido para 250 e 500 indivíduos. Para critério de parada do algoritmo foi considerado o número máximo de 1000 interações. As respostas obtidas foram usadas para a formulação da base de conhecimento *fuzzy* na implementação de modificações no algoritmo ED.

Depois de formulado a base de conhecimento *fuzzy*, modificações foram feitas via lógica *fuzzy* no algoritmo ED visando melhorar sua velocidade de convergência sem comprometer a qualidade de estimação dos parâmetros. O algoritmo ED tem seus parâmetros C_R e F otimizados por ajustador *fuzzy* que atua na evolução dos parâmetros estimados. Nesta etapa, é feita um segundo conjunto de testes de 100 simulações com o algoritmo ED com a melhor configuração de parâmetros C_R , F e N_p obtida da primeira bateria de testes e suas modificações por lógica *fuzzy*. Nas simulações foram avaliados o número de interações, o tempo

de convergência e a validação dos parâmetros estimados. Para critério de parada dos algoritmos foi usado coeficiente de correlação R^2 igual a 0.91 e número de iterações máxima igual a 1000.

Para avaliação da estimação dos modelos matemáticos, o *MSE (Medium of Squared Error)* e o coeficiente de correlação múltipla R^2 são os critérios de desempenho utilizados. Estes valores e os valores estimados do sistema SIMO estão representados por média $\bar{\mu}$ e desvio padrão σ .

Para comparação dos resultados do tempo de convergência do algoritmo ED e suas modificações por lógica *fuzzy*, foram aplicados o teste de hipóteses como método estatístico. O teste de hipótese consiste em comparar o valor de z_t que está associado à hipótese alternativa (H_1) e o valor da estatística do erro de confiança z_α que está associado à hipótese nula (H_0). Para esta comparação é usada a curva de distribuição normal para o cálculo da probabilidade [14].

Para duas amostras populacionais diferentes com valores de média e desvio padrão conhecidos, com nível de confiança de 99%. O valor estatístico teste z_t a ser usada:

$$z_t = \frac{(\bar{x}_1 - \bar{x}_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \quad (4)$$

Onde \bar{x}_1 , σ_1 e n_1 são respectivamente a média, o desvio-padrão e o número de elementos da amostra 1. Enquanto \bar{x}_2 , σ_2 e n_2 são respectivamente a média, o desvio-padrão e o número de elementos da amostra 2 [14].

O teste de hipóteses é realizado com 99% de confiabilidade, logo: $z_\alpha = -2.37$. Para esta condição o teste de hipóteses terá direção unidirecional à esquerda ($z_t < z_\alpha$). Se o valor de z_t for menor que z_α , H_1 é aceita. Se z_t for maior que z_α , H_0 é aceita.

O teste de hipóteses compara dois algoritmos por vez, para H_0 foi considerado o algoritmo com o maior valor médio de tempo de estimação (o mais lento) e para H_1 o algoritmo supostamente que converge mais rápido com o menor valor médio de tempo de estimação (o mais rápido).

Para esta pesquisa foi utilizado como materiais: um computador com processador *intel i7*, 16 Giga bytes de memória e 1 Tera bytes de HD; *software Matlab®*. Os algoritmos de estimação dos sistemas multivariáveis utilizaram dados dos reservatórios fornecidos pela Sanepar®. Os conjuntos de dados utilizados no processo de identificação têm tempo de amostragem igual à uma hora.

4.1 Resultados da Etapa 1.

Na primeira etapa, foram obtidos o melhor modelo de representação do sistema SIMO (TABELA I), os melhores valores de F e C_R fixos e comportamento de evolução do melhor modelo (Fig. 3). O melhor modelo de representação foi o modelo não linear de Hammerstein e seus parâmetros estimados estão representados na TABELA II.

TABELA I: Resultados da identificação do modelo SIMO (1ª Etapa).

Modelo	Algoritmo ED/Best/1/Bin.	Estimação		Validação	
	$C_R/F/N_p/d$	R^2	MSE	R^2	MSE
Linear	0.8/0.6/100/10	0.4168	40.146	-1.116	59.278
Hammerstein	1.0/0.9/500/15	0.9244	5.2014	0.9221	4.60
Wiener	0.9/0.6/500/20	0.6220	26.024	0.6554	20.418
Hammerstein-Wiener	0.9/0.5/500/25	0.6842	21.742	0.5185	28.532

No resultado da primeira etapa (TABELA I), o modelo de Hammerstein modelou melhor o modelo SIMO, enquanto os outros modelos não conseguiram o mesmo desempenho. O modelo de Hammerstein foi escolhido para a segunda etapa para aplicações das modificações *fuzzy* do algoritmo ED.

TABELA II: Parâmetros Obtidos do modelo SIMO Hammerstein (1ª Etapa).

Parâmetros Obtidos					
Coeficientes da matriz A					
a_{11}	a_{21}	a_{31}	a_{12}	a_{22}	a_{32}
-1.8618	1.0577	-0.1672	-1.1073	0.0713	0.0524
Coeficientes do vetor B					
b_0	b_1	b_2	b_3		
-0.0606	0.0760	-0.0234	0.0231		
Não linearidade estática de Hammerstein					
$x(k) = x_1.u(k) + x_2.u(k)^2 + x_3.u(k)^3 + x_4.u(k)^4 + x_5.u(k)^5$					
x_1	x_2	x_3	x_4	x_5	
-9725.8	827.3159	-26.2914	0.3701	-0.0019	

Os ajustadores de parâmetros *fuzzy* foram projetados a partir do comportamento de evolução do algoritmo ED na primeira bateria de testes que teve como objetivo obter os melhores valores de C_R , F e N_p . Este comportamento está expresso por R^2 em função do número de interações e dos valores de C_R e F (Fig. 3).

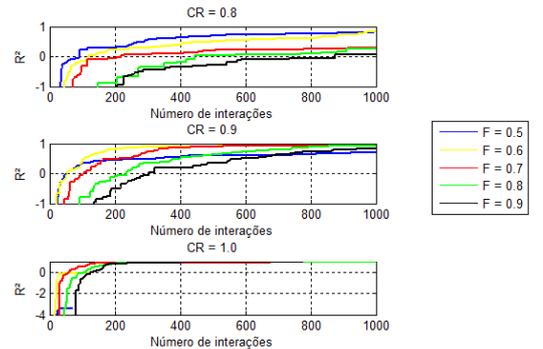


Fig. 3: Gráficos de R^2 em função do número de interações, valores de F e C_R .

Pode-se observar na Fig. 3 que a convergência da evolução dos parâmetros expressa por R^2 é mais rápida quando C_R é igual a 0,9 e F assume os menores valores. Assim, o conjunto de regras (TABELA V) e as funções de pertinências (TABELA III e TABELA IV) foram baseadas nesse comportamento.

A melhor configuração obtida na bateria para os parâmetros do algoritmo ED são: $C_R = 0.9$, $F = 0.6$ e $N_p = 500$.

TABELA III: Funções de pertinências do ajuste do algoritmo ED por MOM.

Função de pertinência da entrada MoM			
Funções de pertinência	Limite inferior	Valor de Pico	Limite superior
IN	-	0	0.25
IM	0	0.25	0.5
ME	0.25	0.5	0.75
MF	0.5	0.75	1.0
FM	0.75	1.0	-
Função de pertinência da saída C_R			
Funções de pertinência	Limite inferior	Valor de Pico	Limite superior
PQ	-	0.9	0.925
PM	0.9	0.925	0.95
MD	0.925	0.95	0.975
MG	0.95	0.975	1.0
GD	0.975	1.0	-
Função de pertinência da saída F			
Funções de pertinência	Limite inferior	Valor de Pico	Limite superior
PQ	-	0.6	0.65
PM	0.6	0.65	0.7
MD	0.65	0.7	0.75
MG	0.7	0.75	0.8
GD	0.75	0.8	-

TABELA IV: Funções de pertinências do ajuste do algoritmo ED por IR.

Função de pertinência da entrada IR			
Funções de pertinência	Limite inferior	Valor de Pico	Limite superior
BA	-	0	0.25
BM	0	0.25	0.5
MD	0.25	0.5	0.75
MA	0.5	0.75	1.0
AL	0.75	1.0	-
Função de pertinência da saída C_R			
Funções de pertinência	Limite inferior	Valor de Pico	Limite superior
PQ	-	0.9	0.925
PM	0.9	0.925	0.95
MD	0.925	0.95	0.975
MG	0.95	0.975	1.0
GD	0.975	1.0	-
Função de pertinência da saída F			
Funções de pertinência	Limite inferior	Valor de Pico	Limite superior
PQ	-	0.675	0.7
PM	0.675	0.7	0.725
MD	0.7	0.725	0.75
MG	0.725	0.75	0.775
GD	0.75	0.775	-

TABELA V: Conjunto de Regras.

Inferência para IR	
Se (IR é AL)	Então (C_R é PQ e F é PQ)
Se (IR é MA)	Então (C_R é PM e F é PM)
Se (IR é MD)	Então (C_R é MD e F é MD)
Se (IR é BM)	Então (C_R é MG e F é MG)
Se (IR é BA)	Então (C_R é GD e F é GD)
Inferência para MoM	
Se (MoM é IN)	Então (C_R é PQ e F é PQ)
Se (MoM é IM)	Então (C_R é PM e F é PM)
Se (MoM é ME)	Então (C_R é MD e F é MD)
Se (MoM é MF)	Então (C_R é MG e F é MG)
Se (MoM é FM)	Então (C_R é GD e F é GD)

4.2 Resultados das Modificações Fuzzy no Algoritmo ED (Etapa 2).

Na segunda etapa foram realizadas as modificações fuzzy na melhor configuração do algoritmo ED, que se obteve na primeira etapa de testes.

TABELA VI: Parâmetros médios estimados para 100 estimativas do modelo SIMO de Hammerstein.

Parâmetros estimados	Algoritmos				
	ED	$Fuzzy MoM$	$Fuzzy IR$	$Fuzzy MoM e IR$	
a_{11}	$\bar{\mu}$	-2.1397	-2.1518	-2.1521	-2.1732
	σ	0.1695	0.1538	0.1677	0.1531
a_{21}	$\bar{\mu}$	1.5468	1.5669	1.5681	1.6032
	σ	0.2820	0.2568	0.2789	0.2555
a_{31}	$\bar{\mu}$	-0.3864	-0.3947	-0.3958	-0.4105
	σ	0.1191	0.1091	0.1176	0.1086
a_{12}	$\bar{\mu}$	-1.1997	-1.2060	-1.2045	-1.2115
	σ	0.0633	0.0576	0.0628	0.0562
a_{22}	$\bar{\mu}$	0.2058	0.2136	0.2116	0.2212
	σ	0.0894	0.0832	0.0871	0.0784
a_{32}	$\bar{\mu}$	0.0058	0.0041	0.0045	0.0016
	σ	0.0325	0.0307	0.00294	0.0270
b_0	$\bar{\mu}$	-0.0178	-0.0160	-0.0156	-0.0120
	σ	0.0318	0.0284	0.0308	0.0280
b_1	$\bar{\mu}$	0.0228	0.0216	0.0198	0.0152
	σ	0.0398	0.0376	0.0383	0.0347
b_2	$\bar{\mu}$	-0.0089	-0.0090	-0.0077	-0.0058
	σ	0.0150	0.0155	0.0143	0.0124
b_3	$\bar{\mu}$	0.0088	0.0073	0.0078	0.0060
	σ	0.0160	0.0132	0.0159	0.0136
x_1	$\bar{\mu}$	1.3681e4	6.9245e4	7479.5	1.3258e4
	σ	3.5075e4	5.0423e5	1.1061e5	4.8923e4
x_2	$\bar{\mu}$	-811.62	-4652	-657.8276	-1035.4
	σ	3099.8	3.2343e4	6867.6	2950.5
x_3	$\bar{\mu}$	13.4326	104.2337	21.6692	29.6639
	σ	149.3305	656.4984	131.3401	52.1336
x_4	$\bar{\mu}$	-4.115e-4	-0.8004	-0.3167	-0.3681
	σ	3.1499	3.5777	0.7841	0.2661
x_5	$\bar{\mu}$	-0.0011	4.4587e-4	0.0017	0.0017
	σ	0.0230	0.0137	0.0073	0.0039

TABELA VII: Número de interações, tempo, MSE estimação, R^2 estimação, MSE validação, e R^2 validação para 100 estimativas do modelo SIMO de Hammerstein.

Dados obtidos	Algoritmos				
	ED	$Fuzzy MoM$	$Fuzzy IR$	$Fuzzy MoM e IR$	
Número de interações	$\bar{\mu}$	488.94	279.12	260.42	189.09
	σ	190.8471	55.6790	60.1765	40.8516
Tempo (s)	$\bar{\mu}$	437.2846	264.8453	246.7001	194.2036
	σ	176.7621	55.4065	56.1809	42.6025
MSE (Estimação)	$\bar{\mu}$	6.2028	6.1774	6.1837	6.1855
	σ	0.1594	0.0204	0.0202	0.0110
R^2 (Estimação)	$\bar{\mu}$	0.9099	0.9103	0.9102	0.9102
	σ	0.0023	2.964e-4	2.929e-4	1.596e-4
R^2 (melhor modelo da Estimação)		0.9117	0.9114	0.9121	0.9107
MSE (Validação)	$\bar{\mu}$	5.5236	5.3778	5.4152	5.3884
	σ	0.4218	0.3239	0.3370	0.2948
R^2 (Validação)	$\bar{\mu}$	0.9068	0.9092	0.9086	0.9091
	σ	0.0071	0.0055	0.0057	0.0050

TABELA VIII: Teste de Hipóteses para o tempo de estimação para 100 estimativas do modelo SIMO de Hammerstein.

Teste unilateral à esquerda, com confiabilidade de 99%.			
Se: $z_t \geq z_{\alpha}$, logo aceita H_0 e rejeita H_1 .			
Se: $z_t < z_{\alpha}$, logo aceita H_1 e rejeita H_0 .			
Comparação entre algoritmos	z_t	$H_0: P(z_t \geq z_{\alpha})$	$H_1: P(z_t < z_{\alpha})$
H0: ED H1: ED <i>fuzzy</i> MOM	-9.3089	Rejeitado	Aceito
H0: ED H1: ED <i>fuzzy</i> IR	-10.2754	Rejeitado	Aceito
H0: ED H1: ED <i>fuzzy</i> MOM e IR	-13.3690	Rejeitado	Aceito
H0: ED <i>fuzzy</i> MOM H1: ED <i>fuzzy</i> IR	-2.2995	Aceito	Rejeitado
H0: ED <i>fuzzy</i> MOM H1: ED <i>fuzzy</i> MOM e IR	-10.1073	Rejeitado	Aceito
H0: ED <i>fuzzy</i> IR H1: ED <i>fuzzy</i> MOM e IR	-7.4455	Rejeitado	Aceito

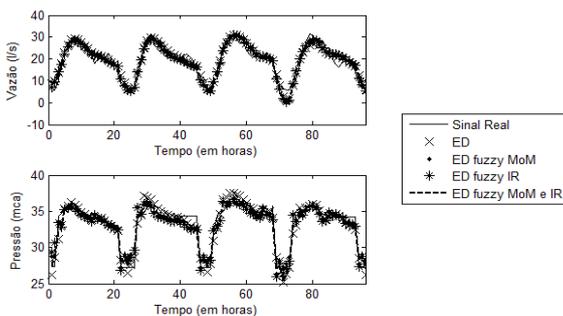


Fig. 4: Resultado da estimação do sistema SIMO a partir do melhor valor de R^2 da estimação de cada algoritmo.

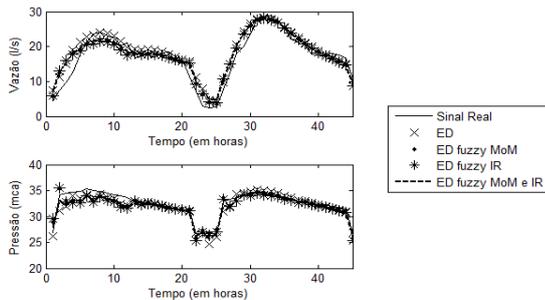


Fig. 5: Resultado da validação do sistema SIMO a partir do melhor valor de R^2 da estimação de cada algoritmo.

Pode ser observado nas tabelas dos parâmetros estimados e de desempenho que os ajustadores *fuzzy* deixaram o processo de estimação do algoritmo ED mais rápido e sem comprometer a convergência dos parâmetros. Isto é confirmado pelo teste de hipóteses (TABELA VIII). Observa-se na TABELA VI que os valores de média e desvio-padrão dos parâmetros estimados do sistema SIMO de todos os algoritmos são muito próximo.

Os resultados do teste de hipóteses mostram que os algoritmos modificados por lógica *fuzzy* (H_1) melhoram o algoritmo ED (H_0) em relação ao tempo de estimação. O teste de hipóteses também mostra que o algoritmo ED que usa a combinação das estratégias MOM e IR converge mais rápido

em relação aos algoritmos ED que usam as estratégias IR e MOM de forma isolada.

A combinação das estratégias MoM e IR melhorou ainda mais a velocidade de convergência do algoritmo em relação ao número de interações, mesmo aumentando o esforço computacional a cada interação. A partir das melhores estimativas (melhor valor de R^2 na TABELA VII), as Fig. 4 e 5 mostram que as modificações *fuzzy* não comprometeram a qualidade do processo de estimação e validação em relação ao algoritmo ED.

CONCLUSÃO

Este trabalho teve como objetivo a aplicação de modificações no processo de cruzamento e mutação do algoritmo ED visando diminuir o tempo de estimação de parâmetros de um sistema SIMO. As modificações baseadas em lógica *fuzzy* melhoraram o desempenho do algoritmo ED na sua versão *ED/Best/1/Bin* em relação ao esforço computacional (tempo total de processamento) sem afetar a qualidade dos parâmetros estimados.

Vislumbra-se para o sistema SIMO real utilizado neste trabalho, que o algoritmo ED pode ser implementado em sistemas de controle inteligentes ou preditivo que utilize um estimador de planta em sistemas reais não lineares cuja dinâmica varia. Logo, as modificações baseadas em lógica *fuzzy* podem melhorar a eficiência de controladores inteligentes ou preditivos.

REFERÊNCIAS

- [1] G. Goodwin, S. F. Graebe and M. E. Salgado, Control System Design, Valparaiso, Chile, 2000.
- [2] S. A. Billings, Nonlinear System Identification, Chichester, UK: John Wiley & Sons, Ltd, 2013.
- [3] S. X. Yang, Nature-Inspired Metaheuristic Algorithms, 2º ed, UK: Luniver Press, 2010.
- [4] C. Lee & X. Lao, "Based on the Lévy Probability Distribution", IEEE Transactions on Evolutionary Computation, v.8, n.1, p. 1-13, 2004.
- [5] K. Price, R. M. Storn and J. A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer, 2005.
- [6] G. Vasu, K. V. S. Santosh and G. Sandeep, "Reduction of large scale linear dynamic SISO and MIMO systems using Differential Evolution Optimization Algorithm". IEEE Students Conference on Electrical, Electronics and Computer Science, p. 1-6, 2012.
- [7] B. Filipic & I. Lorencin, "Evolutionary Multiobjective Design of Alternative Energy Supply System". IEEE World Congress on Computational Intelligence, p. 10-15, 2012.
- [8] J. Zhang & A. C. Sanderson, "JADE: Adaptive Differential Evolution with Optional External Archive". IEEE Transactions on Evolutionary Computation, p. 1-14, 2009.
- [9] B. Subudhi & D. Jena, "A differential evolution based neural network approach to nonlinear system identification", Journal: Applied Soft Computing, Elsevier v. 11, n. 1, p. 861-871, 2011.
- [10] L. A. Zadeh, "Fuzzy Sets", Information and Control, p. 338-353, 1965.
- [11] V. H. A. Marques, Adaptação de Parâmetros em Meta-heurísticas com Sistemas Nebulosos Genéticos. Dissertação de Mestrado. Campinas, Universidade de Campinas, São Paulo, 2011.
- [12] SANEPAR, Dados dos Reservatórios Tarumã e São Braz, 2014.
- [13] J. E. S. Santos, Controle Preditivo não linear para Sistemas de Hammerstein. Tese de Doutorado. Florianópolis, UFSC, 2003.
- [14] D. C. Montgomery, & G. C. Runger, (2012). Estatística Aplicada e Probabilidade para Engenheiros. 5º ed. LTC, 2012.