

Estudo e Implementação de Sistema Lógico Fuzzy na Evolução Diferencial

Bernardo C. P. Nascimento¹
IMECC, Campinas, SP

Resumo. Este projeto visa ao estudo e à implementação de um Sistema Lógico Fuzzy (SLF) para autorregulação dos parâmetros de otimização do algoritmo de otimização global Evolução Diferencial (ED) - nomeado como variante FaDE (*Fuzzy Adaptive Differential Evolution*), e introduzido pela primeira vez no artigo de Liu e Lampinen [1]. A saber, busca-se adaptar os valores da taxa de mutação e a constante de *crossover*, adequando-os às necessidades de maior exploração ou reconhecimento do espaço de busca a cada estágio de otimização. Os programas foram escritos em Python, e podem ser acessados pelo repositório do projeto (indicado na seção **Implementação Computacional**), assim como as animações que, com logro, ilustram as disparidades entre cada variante do ED.

Palavras-chave. Otimização global, Sistema Lógico Fuzzy.

1 Introdução

A fim de contornar as fragilidades inerentes aos algoritmos de otimização local - como a sensibilidade a pontos estacionários locais, e dependência de ordens de continuidade no espaço de busca - surge a necessidade de formulação de técnicas disruptivas. Assim, despontam as meta-heurísticas de busca e otimização global. Uma classe de heurística reconhecidamente eficiente consiste na Evolução Diferencial (ED) - originalmente formulada por Storn e Price [2] -, a qual, inspirando-se na lógica evolutiva darwiniana, continuamente seleciona um conjunto de “indivíduos” (tidos como pontos de amostra da função objetivo) em direção ao “estado evolucionário ótimo” (ponto de mínimo ou máximo global). Daí, unindo processos estocásticos de “mutação” (alteração dos pontos de amostra) e “reprodução” (união de informações de pontos para a geração de um novo), a ED visa a reprodução de dinâmicas da natureza. Contudo, ela exige a passagem de parâmetros pelo usuário, determinantes para a performance do processo de busca. Diante de tal impasse, surgem variantes do ED com processos adaptativos de seleção de parâmetros - performadas em tempo de execução -, e regidas por alguma espécie de inteligência computacional. Logo, a heurística adquire maior autonomia, adequando o valor dos parâmetros aos diferentes estágios da otimização. É nesse contexto que os Sistemas Lógicos Fuzzy (SLF) surgem como alternativa de inteligência computacional: partindo da Inferência de Mamdani para ajustar os parâmetros que regem a taxa de mutação, e reprodução dos indivíduos. O projeto busca o estudo, aprofundamento e implementação das técnicas Fuzzy relativas à ED.

2 Modelo Matemático

2.1 Evolução Diferencial

O algoritmo de ED age segundo uma dinâmica populacional evolutiva ao longo das gerações. Sob esse raciocínio, pode-se conceber cada geração como o conjunto de candidatos a ponto ótimo a

¹b245837@dac.unicamp.br

cada iteraç o do algoritmo. Supondo uma funç o objetivo $f : \mathbb{R}^n \rightarrow \mathbb{R}$, tem-se que uma populaç o de n mero G   formada pelos vetores $x_{i,G} \in \mathbb{R}^n$, com $i = 1, 2, \dots, NP$, em que NP   o n mero fixo de indiv duos para toda geraç o. Dessa maneira, pode-se dividir o algoritmo em quatro etapas: inicializaç o, mutaç o, recombinaç o e seleç o. A otimizaç o finaliza uma vez que se alcança algum crit rio de satisfaç o (como o valor da funç o objetivo), estagnaç o (norma do res duo entre indiv duos  timos de geraç es consecutivas), ou de disp ndio computacional (n mero m ximo de iteraç es). A seguir, s o descritos quatro est gios da variante cl ssica de ED, a **rand-1-bin**.

Inicializaç o

Nesta etapa, gera-se a populaç o inicial de indiv duos. Assumindo integral desconhecimento do espaço de busca, conv m inicializar os vetores com distribuiç o uniforme ao longo de todo o espaço. Da , para a geraç o $G = 0$, s o criados os vetores $x_{i,0}$, $i = 1, 2, \dots, NP$, tais que $x_{i,j,0} \in [x_j^I, x_j^S]$, $j = 1, 2, \dots, n$ - intervalo de factibilidade para cada entrada j do vetor n -dimensional.

Mutaç o

Neste est gio,   criado um vetor mutante para cada indiv duo mediante a combinaç o linear das entradas entre membros da populaç o atual, a fim de expandir a  rea de busca abarcada pela populaç o. Em suma, para todo indiv duo alvo $x_{i,G}$, $i = 1, 2, \dots, NP$, define-se o vetor mutante conforme a estrat gia de mutaç o pr -selecionada (mais um hiperpar metro associado   t cnica). Classicamente, define-se o mutante segundo a heur stica denominada **rand-1**, dada por:

$$v_{i,G+1} = x_{r_1,G} + F \cdot (x_{r_2,G} - x_{r_3,G}) \quad (1)$$

onde $r_1, r_2, r_3 \in \{1, 2, \dots, NP\} \setminus \{i\}$, sendo todos eles distintos, e $F \in [0, 2]$   o par metro real de mutaç o - posteriormente regulado pelo SLF. O papel do par metro F consiste, justamente, em amplificar, ou amortizar, a tend ncia de variabilidade dos vetores $x_{r_2,G}$ e $x_{r_3,G}$ sobre o vetor mutante final. Em relaç o ao nome da pol tica de mutaç o, ele se justifica por selecionar aleatoriamente (**rand**) um indiv duo base, no caso sendo $x_{r_1,G}$; e lançar m o de t o somente um (1) vetor de diferenças, este associado aos vetores $x_{r_2,G}$ e $x_{r_3,G}$.

Recombinaç o

Aqui, incorpora-se a diversidade de soluç es oriundas dos vetores de mutaç o   populaç o atual. Matematicamente, constr i-se um indiv duo teste $u_{i,G+1}$ cujas entradas s o definidas conforme o esquema **bin** (nomenclatura atribuída devido   aplicaç o de experimentos binomiais independentes):

$$u_{i,j,G+1} = \begin{cases} v_{i,j,G+1} & , \text{ se } randb(j) \leq CR, \text{ ou } j = rnbr(i) \\ x_{i,j,G} & , \text{ sen o} \end{cases} \quad , \quad j = 1, 2, \dots, n. \quad (2)$$

Na equa  o (2), $randb(j)$ representa o j - simo sorteio de um n mero uniformemente distribuído no intervalo $[0, 1]$, enquanto $CR \in [0, 1]$   a constante de *crossover* (ou permuta  o gen tica) - que tamb m vir  a ser regulada pelo SLF. J  $rnbr(i) \in \{1, 2, \dots, n\}$ representa um  ndice aleatoriamente escolhido - crit rio que garante a presen a de ao menos um gene oriundo da muta  o, de modo a evitar estagna  o prematura dos vetores teste.

Sele  o

Por fim, cada indiv duo alvo $x_{i,G}$   comparado com seu filho (isto  , seu vetor teste correspondente) $u_{i,G+1}$, e seleciona-se para a pr xima gera  o aquele que, avaliado por f , apresenta o menor valor - considerando o problema de minimiza  o. Em resumo, determina-se $x_{i,G+1}$, $i = 1, 2, \dots, NP$, de acordo com

$$x_{i,j,G+1} = \begin{cases} x_{i,j,G} & , \text{ se } f(x_{i,G}) \leq f(u_{i,G+1}) \\ u_{i,j,G+1} & , \text{ sen o} \end{cases} \quad (3)$$

2.2 Sistema Lógico Fuzzy

Do ponto de vista da inteligência computacional, um SLF destaca-se pela capacidade de aproximação universal, bem como pela fácil implementação computacional (Valle, 2025 [3]). Daí, qualquer Sistema Fuzzy pode ser definido por quatro componentes:

- Dicionário: conjuntos fuzzy sobre as variáveis;
- Base de regras: relações de implicação fuzzy entre as variáveis;
- Método de inferência: esquema de extração de uma resposta fuzzy dada um entrada qualquer - neste presente projeto, bem como no artigo original, lançou-se mão do Método de Mamdani;
- Desfuzzificação: heurística de conversão de saída fuzzy em um número real.

Sendo assim, considere um sistema que receba como entrada um vetor $u = [u_1, \dots, u_n] \in \mathbf{U}$, retorne uma saída unidimensional $v \in \mathbf{V}$, e tenha k regras definidas como

$$\text{Se } u \text{ é } A_i \text{ então } v \text{ é } B_i \quad \forall i = 1, 2, \dots, k \quad (4)$$

em que $A_i = \mathcal{F}(A_{i1} \times A_{i2} \times \dots \times A_{in})$. Daí, em consonância com o método de Mamdani, definem-se os pesos de ativação de cada regra i por $w_i = A_i(u) = \bigwedge_{j=1}^n A_{ij}(u_j)$. Em seguida, pode-se determinar a saída fuzzy por $B(v) = \bigvee_{i=1}^k [w_i \wedge B_i(v)]$. Formalmente, a inferência em análise define uma relação fuzzy $\mathcal{M} \in \mathcal{F}(\mathbf{U} \times \mathbf{V})$ que associa a cada valor de entrada u , uma saída correspondente v , cuja função de pertinência é dada por

$$\mathcal{M}(u, v) = \max_{1 \leq i \leq k} [A_i(u) \wedge B_i(v)] \quad (5)$$

Visto que emprega a t-norma do mínimo como conjunção, e t-conorma do máximo como disjunção, a inferência de Mamdani também é referida como inferência max-min.

Por fim, o método inclui a desfuzzificação da resposta via centroide da função de pertinência $B(v)$ definida, isto é, extrai-se \hat{v} através de

$$\hat{v} = \frac{\sum_{i=1}^n v_i \cdot B(v_i)}{\sum_{i=1}^n B(v_i)} \quad (6)$$

2.3 FaDE

Uma vez munidos de amparo teórico sobre SLF's, pode-se avaliar um plano de ação para regular, em tempo de execução da ED, certos hiperparâmetros. Ao longo da evolução das gerações de candidatos à solução, sabe-se que cada indivíduo só é substituído em caso de superioridade, em termos da função objetivo, do seu respectivo filho.

Com isso em mente, pode-se relacionar, à variação das entradas de cada indivíduo entre duas gerações consecutivas, a tendência de melhora do custo à medida que se explora mais o espaço de busca ao redor (caso este em que os indivíduos consecutivos alteram muito seu valor), ou ao passo que se reconhece minuciosamente a região já populada (caso de baixa variabilidade entre o valor das entradas de duas gerações seguidas). Aliada à magnitude de variação das entradas dos indivíduos, a análise sobre a variação do valor da função objetivo para cada indivíduo, entre gerações consecutivas, reforça as evidências sobre a tendência populacional durante a otimização.

Matematicamente, é possível definir os valores acima como PC (relativo à mudança do valor dos indivíduos) e FC (relacionado à variação dos valores da função objetivo) dados, conforme Liu e Lampinen, pela raiz quadrada dos desvios médios:

$$PC = \sqrt{\frac{1}{NP} \sum_{i=1}^{NP} \|x_{i,G+1} - x_{i,G}\|_2^2} \quad , \quad (7)$$

e

$$FC = \sqrt{\frac{1}{NP} \sum_{i=1}^{NP} [f(x_{i,G+1}) - f(x_{i,G})]^2} \quad . \quad (8)$$

A partir de manipulações de escala sobre essas grandezas, é possível alimentar dois Sistemas Lógicos Fuzzy distintos, sendo um deles voltado à regulação da constante de mutação F , e outro da constante de permutação genética CR .

Para o Sistema de inferência de F , definem-se as entradas $f_1 = 1 - (1 + PC) \cdot e^{-PC}$, $f_2 = 1 - (1 + FC) \cdot e^{-FC}$, que irão alimentar a aplicação $\mathcal{S}_F \in \mathcal{F}(\mathbb{R} \times \mathbb{R} \times \mathbb{R})$. O sistema de regras, inspirado nos cenários supracitados de tendência exploratória ou de reconhecimento, podem ser construídos como segue:

f_1	\wedge	f_2	\rightarrow	v
S_{f_1}		S_{f_2}		S_F
S_{f_1}		M_{f_2}		M_F
S_{f_1}		B_{f_2}		B_F
M_{f_1}		S_{f_2}		M_F
M_{f_1}		M_{f_2}		M_F
M_{f_1}		B_{f_2}		B_F
B_{f_1}		S_{f_2}		B_F
B_{f_1}		M_{f_2}		B_F
B_{f_1}		B_{f_2}		B_F

Tabela 1: Sistema de regras para inferir F

c_1	\wedge	c_2	\rightarrow	v
S_{c_1}		S_{c_2}		S_{CR}
S_{c_1}		M_{c_2}		M_{CR}
S_{c_1}		B_{c_2}		B_{CR}
M_{c_1}		S_{c_2}		M_{CR}
M_{c_1}		M_{c_2}		M_{CR}
M_{c_1}		B_{c_2}		B_{CR}
B_{c_1}		S_{c_2}		B_{CR}
B_{c_1}		M_{c_2}		B_{CR}
B_{c_1}		B_{c_2}		B_{CR}

Tabela 2: Sistema de regras para inferir CR

em que S, M e B correspondem aos números fuzzy associados às variações de magnitudes *pequenas*, *médias* e *grandes* respectivamente. No artigo original, os conjuntos foram definidos como números fuzzy gaussianos, isto é, $G(\sigma, \mu) : \mathbb{R} \rightarrow [0, 1]$, com

$$G(\sigma, \mu)(x) = \exp\left[-\frac{(x - \mu)^2}{2\sigma^2}\right] \quad . \quad (9)$$

Nesse primeiro sistema, tem-se que $S_{f_1} = G(0, 25, 0, 05)$, $M_{f_1} = G(0, 25, 0, 5)$, $B_{f_1} = G(0, 25, 0, 9)$, $S_{f_2} = G(0, 35, 0, 01)$, $M_{f_2} = G(0, 35, 0, 5)$, $B_{f_2} = G(0, 35, 0, 9)$, e $S_F = G(0, 5, 0, 3)$, $M_F = G(0, 5, 0, 6)$, $B_F = G(0, 5, 0, 9)$.

Já para o sistema de inferência de CR , os autores definem as entradas $c_1 = 2 \cdot f_1$ e $c_2 = 2 \cdot f_2$, que alimentam a aplicação $\mathcal{S}_{CR} \in \mathcal{F}(\mathbb{R} \times \mathbb{R} \times \mathbb{R})$. Analogamente ao primeiro SLF, define-se o mesmo conjunto de regras, onde $S_{c_1} = G(0, 5, 0, 1)$, $M_{c_1} = G(0, 5, 0, 8)$, $B_{c_1} = G(0, 5, 1, 5)$, $S_{c_2} = G(0, 5, 0, 1)$, $M_{c_2} = G(0, 5, 0, 8)$, $B_{c_2} = G(0, 5, 1, 5)$, e $S_{CR} = G(0, 35, 0, 4)$, $M_{CR} = G(0, 35, 0, 7)$, $B_{CR} = G(0, 35, 1)$.

3 Experimentos Computacionais

A seguir, são descritos os experimentos numéricos realizados. Seus respectivos resultados são exibidos, bem como análises são tecidas.

3.1 Funções Teste com Alta Dimensionalidade

A princípio, submeteu-se a variante Fuzzy da ED a diferentes funções voltadas ao teste de métodos de otimização. Assim, sua performance (isto é, o ponto de mínimo encontrado) foi comparada à formulação original do algoritmo. No que tange à inicialização dos métodos, ambos partiram da mesma configuração inicial de população, constante de mutação $F = 0,5$ e de *crossover* $CR = 0,9$, e estratégias - aplicou-se tanto **rand-1-bin**, quanto **best-1-bin**, esta última que substitui $x_{r1,G}$ na equação (1), pelo melhor indivíduo da geração G . Foram toleradas $20 \cdot D$ gerações com $10 \cdot D$ indivíduos, em que D representa a dimensão da entrada da função teste $f : \mathbb{R}^D \rightarrow \mathbb{R}$.

Tabela 3: Pontos de mínimo para diferentes funções teste. Em caixas, as melhores aproximações de mínimo global

$f(\cdot)$	$f(\mathbf{x}^*)$	FaDE rand	FaDE best	DE rand	DE best
Rosenbrock (20 D)	0	146,2	20, 2	210,0	138,5
Styblinski-Tang (10 D)	-391,6599	-390,5	-391, 6	-391, 6	-391, 6
Michalewicz (10 D)	-9,66015	-9,2	-9, 5	-8,3	-9,0
Griewank (20 D)	0	21,4	1, 4	145,6	157,3
Ackley (20 D)	0	11,5	0, 3	18,4	18,6

A partir da Tabela 3, evidencia-se a superioridade da variante Fuzzy sobre o conjunto de funções teste selecionado, mais especificamente FaDE munido da estratégia de mutação que tendencia a evolução das gerações em direção do indivíduo mais apto por iteração. Ademais, outro importante fator para análise da técnica é o seu custo computacional, em termos de tempo de execução. Com base na Tabela 5, nota-se que a implementação dos SLF pouco afetou o tempo de execução total das rodadas de otimização para as respectivas estratégias de mutação, visto que as disparidades em tempo restringiram-se aos décimos de segundo. Contudo, vale ressaltar que os testes foram realizados serialmente, de modo que primeiro aplicou-se a minimização por FaDE **rand**, então **best**, e daí DE **rand**, **best** - ordem que pode ter penalizado os últimos métodos.

Tabela 4: Tempos de otimização para diferentes funções teste. Em caixas, os tempos de execução das melhores aproximações de mínimo global

$f(\cdot)$	FaDE rand	FaDE best	DE rand	DE best
Rosenbrock (20 D)	7,2 s	17, 4s	7,3 s	17,3 s
Styblinski-Tang (10 D)	1,3 s	2, 9s	1, 2s	2, 8s
Michalewicz (10 D)	1,8 s	3, 7s	1,8 s	3,8 s
Griewank (20 D)	11,6 s	27, 3s	11,5 s	27,1 s
Ackley (20 D)	8,9 s	21, 7s	10,0 s	23,0 s

Complementarmente, também foram registrados vídeos do processo de minimização de algumas funções teste, comparando a evolução por *DE* e *FaDE*, além das variações dos parâmetros F e CR em tempo de execução. Os vídeos - presentes no repositório indicado na ulterior seção

Implementação Computacional - ilustram bem a disparidade estratégica entre as variantes, jogando luz à inteligência computacional em razão dos SLF's, sobretudo em momentos de dispersão e exploração coordenada da população ao longo das áreas promissoras. O perfil de comportamento dos parâmetros inferidos também corroboram as tendências codificadas nos sistemas de regra.

3.2 Variação dos Métodos de Desfuzzyficação

Nestes testes, procurou-se variar os métodos de conversão das saídas fuzzy dos SLF's, a fim de avaliar algum incremento na rapidez de convergência do algoritmo FaDE. Para tanto, lançou-se mão de outros dois métodos de desfuzzyficação: média dos máximos, dado por

$$\hat{v} = \frac{\sum_{i \in M} x_i}{\text{Card}(M)} \quad M = \{x \in \mathbf{V} : B(x) = \max_{x \in \mathbf{V}} B(x)\} \quad ; \quad (10)$$

e o centro dos máximos, definido por

$$\hat{v} = \frac{i + s}{2}, \quad i = \inf\{x \in \mathbf{V} : B(x) = \max_{x \in \mathbf{V}} B(x)\} \quad s = \sup\{x \in \mathbf{V} : B(x) = \max_{x \in \mathbf{V}} B(x)\}. \quad (11)$$

Nos moldes do experimento anterior, submeteu-se o FaDE com estratégia **rand-1** à otimização de algumas das funções teste, e registrou-se o histórico do melhor indivíduo de cada geração ao longo de 100 iterações, para cada método de desfuzzyficação citado.

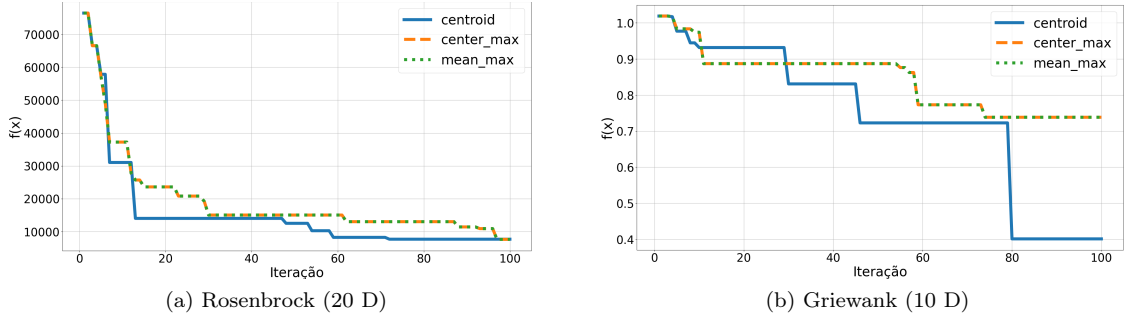


Figura 1: Desempenho do melhor indivíduo a cada geração, para diferentes técnicas de desfuzzyficação. Fonte: imagem autoral.

De modo geral, a técnica do centroide provou-se a mais apropriada, uma vez que manteve-se abaixo das curvas das demais técnicas, ou brevemente foi superada por elas ao longo da otimização. Tais resultados indicam a superioridade de uma abordagem "parcimoniosa" à desfuzzyficação, em detrimento das estratégias mais radicais, como as do máximo, que privilegiam os pontos de supremo da função do consequente.

4 Conclusão

Ao fim desse projeto, atesta-se a satisfatória implementação da técnica proposta por Lampinen e Liu - muito em razão das aplicações computacionais de Sistemas Lógicos Fuzzy disponibilizadas anteriormente pelo professor Marcos Valle.

Para além disso, a avaliação comparativa de performance da ED e sua variante Fuzzy exibiram a superioridade da última no que tange à rapidez de convergência em problemas de alta dimensionalidade. Apesar do custo computacional adicional, oriundo dos SLF's, a variante alcançou tempos de execução similares. Contudo, o primeiro experimento numérico também joga luz sobre a importância das estratégias de mutação e recombinação, hiperparâmetro intrínseco à Evolução Diferencial e, ainda assim, incapaz de ser adaptado pelos sistemas Fuzzy aqui apresentados.

O segundo experimento, apesar de indicar uma superioridade do método do centroide, ilustra como a adição de Lógica Fuzzy sobre a ED acaba por, invariavelmente, incrementar significativamente a quantidade de hiperparâmetros envolvidos. A saber

Implementação computacional

No link a seguir, aquele que lê o presente projeto pode acessar todos os códigos relativos à técnica de compressão, bem como aos experimentos numéricos.

<https://github.com/Beprados/Fuzzy-Logic-Project>

Referências

- [1] J. Lampinen J. Liu. “A Fuzzy Adaptive Differential Evolution Algorithm”. Em: **Soft Computing: A Fusion of Foundations, Methodologies and Applications** (2005). Aceito. DOI: 10.1007/s00500-004-0363-x.
- [2] K. Price R. Storn. “Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces”. Em: **Journal of Global Optimization** (1997). Aceito. DOI: <https://doi.org/10.1023/A:1008202821328>.
- [3] M.E. Valle. **Slides teóricos cedidos em aula**. Online. Acessado em 12/06/2025.