# SELF-ADAPTIVE DIFFERENTIAL EVOLUTION ALGORITHM (SADE)

Cómputo Evolutivo

Arturo Márquez Flores

26 Marzo 2019

Maestría en Inteligencia Artificial Universidad Veracruzana CIIA – Centro de Investigación en Inteligencia Artificial Sebastián Camacho No 5, Xalapa, Ver., México 91000 https://github.com/arturomf94/ce-mia

### CONTENIDO

- · Motivación
- · SaDE
- · Ejemplos

MOTIVACIÓN

### INTRODUCCIÓN

La Evolución Diferencial (DE) depende de parámetros NP, CR y F.

### INTRODUCCIÓN

Existen varias estrategias diferentes para aplicar el operador de mutación:

"DE/rand/1": 
$$\mathbf{V}_{i,G} = \mathbf{X}_{r_1,G} + F \cdot (\mathbf{X}_{r_2,G} - \mathbf{X}_{r_3,G})$$

"DE/best/1":  $\mathbf{V}_{i,G} = \mathbf{X}_{best,G} + F \cdot (\mathbf{X}_{r_1,G} - \mathbf{X}_{r_2,G})$ 

"DE/current to best/1":

 $\mathbf{V}_{i,G} = \mathbf{X}_{i,G} + F \cdot (\mathbf{X}_{best,G} - \mathbf{X}_{i,G}) + F \cdot (\mathbf{X}_{r_1,G} - \mathbf{X}_{r_2,G})$ 

"DE/best/2":

 $\mathbf{V}_{i,G} = \mathbf{X}_{best,G} + F \cdot (\mathbf{X}_{r_1,G} - \mathbf{X}_{r_2,G}) + F \cdot (\mathbf{X}_{r_3,G} - \mathbf{X}_{r_4,G})$ 

"DE/rand/2":

 $\mathbf{V}_{i,G} = \mathbf{X}_{r_1,G} + F \cdot (\mathbf{X}_{r_2,G} - \mathbf{X}_{r_3,G}) + F \cdot (\mathbf{X}_{r_4,G} - \mathbf{X}_{r_5,G})$ 

Figura: Estrategias de Mutación

+

### INTRODUCCIÓN

Para obtener buenos resultados hay que buscar manualmente!

### SADE

#### Self-adaptive Differential Evolution Algorithm for Numerical Optimization

#### A. K. Qin

School of Electrical and Electronic Engineering, Nanyang Technological University 50 Nanyang Ave., Singapore 639798 qinkai@pmail.ntu.edu.sg

#### P. N. Suganthan

School of Electrical and Electronic Engineering, Nanyang Technological University 50 Nanyang Ave., Singapore 639798 epnsugan@ntu.edu.sg

Figura: SaDE

### **IDEA**

La idea principal es seleccionar probabilísticamente una de las estrategias de mutación.

$$\begin{aligned} \mathbf{V}_{i,G} &= \mathbf{X}_{r_1,G} + F \cdot \left( \mathbf{X}_{r_2,G} - \mathbf{X}_{r_3,G} \right) \\ \mathbf{V}_{i,G} &= \mathbf{X}_{i,G} + F \cdot \left( \mathbf{X}_{best,G} - \mathbf{X}_{i,G} \right) + F \cdot \left( \mathbf{X}_{r_1,G} - \mathbf{X}_{r_2,G} \right) \end{aligned}$$

Figura: Estrategias Usadas

#### SADE

- · A cada estrategia se le asigna una probabilidad p<sub>i</sub> de ser elegida.
- · Se inicializa un vector de tamaño NP y a cada entrada se le asigna una estrategia probabilísticamente.
- · Por un período de aprendizaje (50 gens) se acumulan ns<sub>i</sub> y nf<sub>i</sub>.
- · Se actualizan probabilidades.

$$p_1 = \frac{ns1 \cdot (ns2 + nf2)}{ns2 \cdot (ns1 + nf1) + ns1 \cdot (ns2 + nf2)}, \quad p_2 = 1 - p_1$$

Figura: Actualización Probabilidades

#### SADE

- · NP sigue siendo parámetro.
- · F se varía en un intervalo de (0,2], con media 0.5 y d.e. 0.3.
- CR se distribuye normal en un rango con media CRm y d.e. 0.1.
   Se guardan los casos exitosos y después de cierto número de generaciones (25) se recalcula utilizando los casos exitosos.

#### **DESARROLLO:**

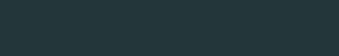
## Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems

Janez Brest, *Member, IEEE*, Sašo Greiner, Borko Bošković, Marjan Mernik, *Member, IEEE*, and Viljem Žumer, *Member, IEEE* 

Figura: Segundo Artículo

$$F_{i,G+1} = \begin{cases} F_l + rand_1 * F_u, & \text{if } rand_2 < \tau_1 \\ F_{i,G}, & \text{otherwise} \end{cases}$$
 
$$CR_{i,G+1} = \begin{cases} rand_3, & \text{if } rand_4 < \tau_2 \\ CR_{i,G}, & \text{otherwise} \end{cases}$$

Figura: Método



**EJEMPLOS** 



En **este colab** podemos experimentar con el algoritmo SaDE.