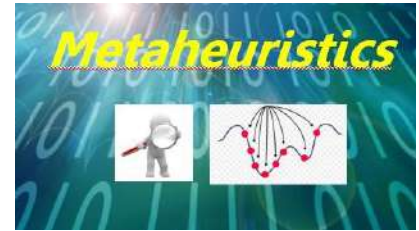


METAHEURÍSTICAS

2022-2023



- Tema 1. Introducción a las Metaheurísticas
- Tema 2. Modelos de Búsqueda: Entornos y Trayectorias vs Poblaciones
- Tema 3. Metaheurísticas Basadas en Poblaciones
- Tema 4: Algoritmos Meméticos
- Tema 5. Metaheurísticas Basadas en Trayectorias
- Tema 6. Metaheurísticas Basadas en Adaptación Social
- Tema 7. Aspectos Avanzados en Metaheurísticas
- Tema 8. Metaheurísticas Paralelas

METAHEURÍSTICAS

TEMA 3. METAHEURÍSTICAS BASADAS EN POBLACIONES

Parte II:

- 1. ALGORITMOS EVOLUTIVOS PARA OPTIMIZACIÓN CONTINUA I (Algoritmos genéticos)**
- 2. EVOLUCIÓN DIFERENCIAL**
- 3. ESTRATEGIAS DE EVOLUCIÓN**
- 4. TEMA 6. PSO. ALGORITMOS DE NUBES DE PARTÍCULAS**
- 5. ALGORITMOS EVOLUTIVOS PARA OPTIMIZACIÓN CONTINUA II (Competiciones y modelos)**
- 6. NUEVOS MODELOS BIOINSPIRADOS PARA OPTIMIZACIÓN DE PARÁMETROS**

METAHEURÍSTICAS

TEMA 3. METAHEURÍSTICAS BASADAS EN POBLACIONES

Parte II:

1. **ALGORITMOS EVOLUTIVOS PARA OPTIMIZACIÓN CONTINUA I (Algoritmos genéticos)**
2. **EVOLUCIÓN DIFERENCIAL**
3. **ESTRATEGIAS DE EVOLUCIÓN**
4. **TEMA 6. PSO. ALGORITMOS DE NUBES DE PARTÍCULAS**
5. **ALGORITMOS EVOLUTIVOS PARA OPTIMIZACIÓN CONTINUA II (Competiciones y modelos)**
6. **NUEVOS MODELOS BIOINSPIRADOS PARA OPTIMIZACIÓN DE PARÁMETROS**

EVOLUCIÓN ARTIFICIAL

Existen cuatro paradigmas básicos:

Algoritmos Genéticos que utilizan operadores genéticos sobre cromosomas. 1975, Michigan University



John Holland
Inventor of genetic algorithms
Professor of CS and Psychology at the U. of Michigan.

Estrategias de Evolución que enfatizan los cambios de comportamiento al nivel de los individuos. 1964, Technische Universität Berlin



Inventors of Evolution Strategies



Ing. Ingo Rechenberg
Bionics & Evolutionstechnik
Technical University Berlin
<http://www.bionik.tu-berlin.de/>

Programación Evolutiva que enfatizan los cambios de comportamiento al nivel de las especies. 1960-1966, Florida



Lawrence J. Fogel,
Natural Selection, Inc.
Inventor of Evolutionary Programming

Programación Genética que evoluciona expresiones representadas como árboles. 1989, Stanford University



John Koza
Stanford University.
Inventor of Genetic Programming

Optimización Evolutiva de Parámetros (Evolutionary Parameter Optimization): Introducción

- Los diseñadores de cada una de las técnicas de Computación Evolutiva vieron que sus problemas particulares podían ser resueltos por algoritmos de evolución.
 - Fogel estaba interesado en resolver programas de evolución.
 - **Rechenberg & Schwefel estaban interesados en resolver problemas de optimización de parámetros.**
 - Holland estaba interesado en el desarrollo de sistemas adaptativos robustos.

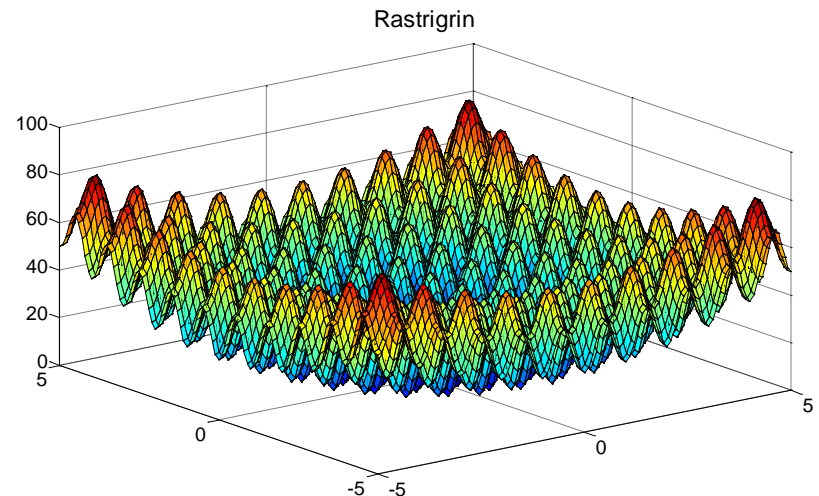
Optimización Evolutiva de Parámetros (Evolutionary Parameter Optimization): Introducción

Centramos nuestra atención en el problema de encontrar el óptimo global de una función con variables reales que está caracterizada por:

**múltiples mínimos
no diferenciable**

$$f(X_i) = D \cdot 10 + \sum_{j=1}^D [x_{ij}^2 - 10 \cdot \cos(2\pi x_{ij})]$$

**Tiene muchos mínimos
locales y es altamente
multimodal.**



Optimización Evolutiva de Parámetros (Evolutionary

Parameter Optimization): Introducción

Motivación del problema

- Hay muchas aplicaciones en las que un científico/ingeniero tiene que optimizar una función no lineal y no diferenciable con múltiples mínimos.

Ejemplo: Problemas en la competición CEC2011

14 problemas



Real-world Numerical Optimization Problems



Special Track: Competition: Testing Evolutionary Algorithms on Real-world Numerical Optimization Problems [CEC'2011](#), New Orleans, USA, Jun 5 - 8, 2011. Organizer: P.N. Suganthan.

4. Special Sessions and Workshops: Problem definitions and contributions (pdf files)
<http://sci2s.ugr.es/EAMHCO/#SS>

1. **Parameter Estimation for Frequency-Modulated (FM) Sound Waves**
2. **Lennard-Jones Potential Problem**
3. **The Bifunctional Catalyst Blend Optimal Control Problem**
4. **Optimal Control of a Non-Linear Stirred Tank Reactor**
5. **Tersoff Potential Function Minimization Problem**
6. **Spread Spectrum Radar Polly phase Code Design**
7. **Transmission Network Expansion Planning (TNEP) Problem**
8. **Large Scale Transmission Pricing Problem**
9. **Circular Antenna Array Design Problem**
10. **Dynamic Economic Dispatch (DED) Problem**
11. **Static Economic Load Dispatch (ELD) Problem**
12. **Hydrothermal Scheduling Problem**
13. **Messenger: Spacecraft Trajectory Optimization Problem**
14. **Cassini 2: Spacecraft Trajectory Optimization Problem**

**13 Algorithms
participate in the
Special Track**

Optimización Evolutiva de Parámetros (Evolutionary

Parameter Optimization): Introducción

Algoritmos Evolutivos de Parámetros Reales Más Populares

- ▶ Algoritmos Genéticos con Codificación Real (*Real-coded (parameter) genetic algorithm* (RCGAs))
- ▶ Estrategias de Evolución (*Evolution strategies* (ES))
- ▶ Evolución Diferencial (*Differential evolution* (DE))
- ▶ Nube de Partículas (*Particle swarm optimization* (PSO))
- ▶ Algoritmos Meméticos con Codificación Real (*Real coding memetic algorithms* (RCMA))

▶ **Recientemente:**

Published: 05 July 2020

Comprehensive Taxonomies of Nature- and Bio-inspired Optimization: Inspiration Versus Algorithmic Behavior, Critical Analysis Recommendations

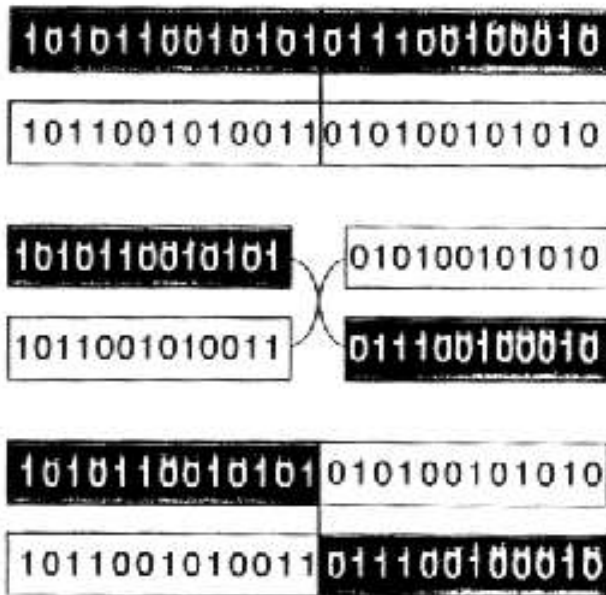
[Daniel Molina](#) , [Javier Poyatos](#), [Javier Del Ser](#), [Salvador García](#), [Amir Hussain](#) & [Francisco Herrera](#)

[Cognitive Computation](#) **12**, 897–939(2020) | [Cite this article](#)

6 Citations | 0 Altmetric | [Metrics](#)

Trabajos pioneros y destacados

Codificación binaria



CROSSOVER is the fundamental mechanism of genetic rearrangement for both real organisms and genetic algorithms.

Chromosomes line up and then swap the portions of their genetic code beyond the crossover point.

**Operador de cruce con parámetros reales.
¿Rediseñar el operador de cruce?**

Algoritmos Genéticos con Codificación Real

- ▶ Las variables de decisión son codificadas directamente, en lugar de usar cadenas binarias.
- ▶ La **recombinación** y **mutación** necesitan cambios estructurales

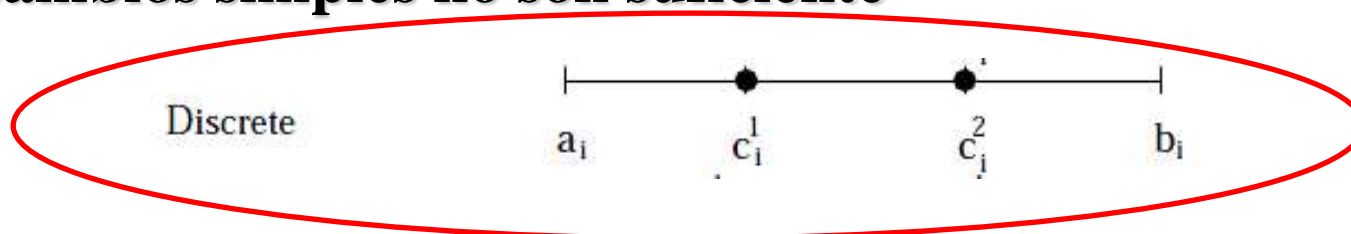
Recombinación

$\Rightarrow ?$

Mutación

$(x_1 x_2 \dots x_n) \Rightarrow ?$

- ▶ El operador de selección sigue siendo el mismo
- ▶ Intercambios simples no son suficiente



Pioneros y trabajos destacados

Problemas con cruce real: Vecindario y Cruce

Idea del cruce: combinar genotipos paternos para que los genotipos de los hijos queden “en algún lugar entre ellos”.

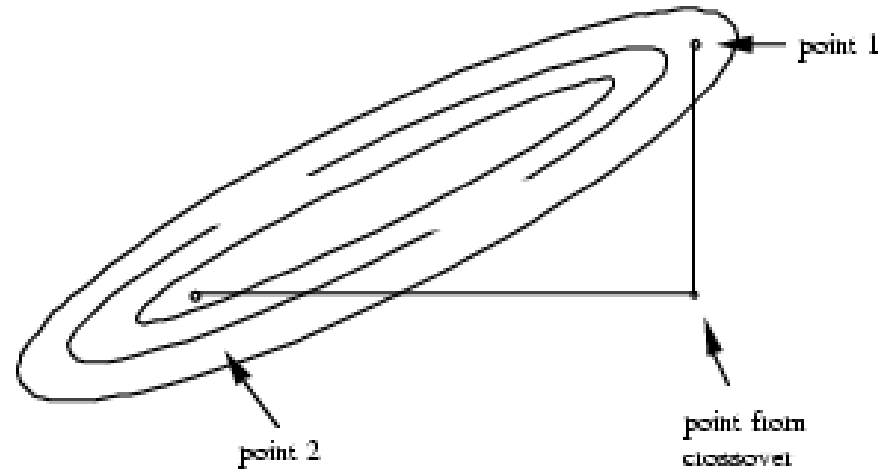


Figure 2

Interpretación y Generalización

El **cruce** y **mutación** tradicional tienen una interpretación natural en la estructura del vecindario en términos de **proximidad** y **intermediación**.

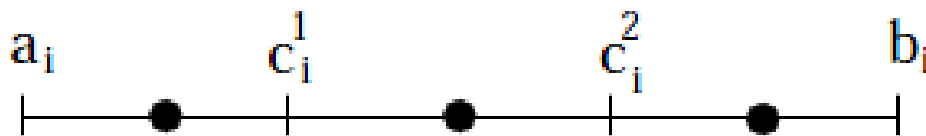
Pioneros y trabajos destacados

Primera propuesta de Codificación Real: Cruce Lineal/Aritmético

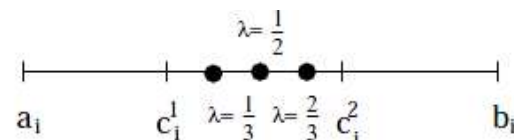
Wright, A. (1991). Genetic Algorithms for Real Parameter Optimization. FOGA 1, 205-218.

• *Cruce lineal*

- *De 2 puntos padres, se generan 3 puntos nuevos:*
 - $(1/2)p1 + (1/2)p2$, $(3/2)p1 - (1/2)p2$, $(-1/2)p1 + (3/2)p2$
 - $(1/2)p1 + (1/2)p2$ es el punto medio de $p1$ y $p2$
 - Los otros están en la línea determinada por $p1$ y $p2$
- *Los 2 de los 3 mejores puntos son enviados a la siguiente generación.*
- *Desventaja- Esquema sumamente alterado. No es compatible con el teorema del esquema descrito en la siguiente diapositiva.*



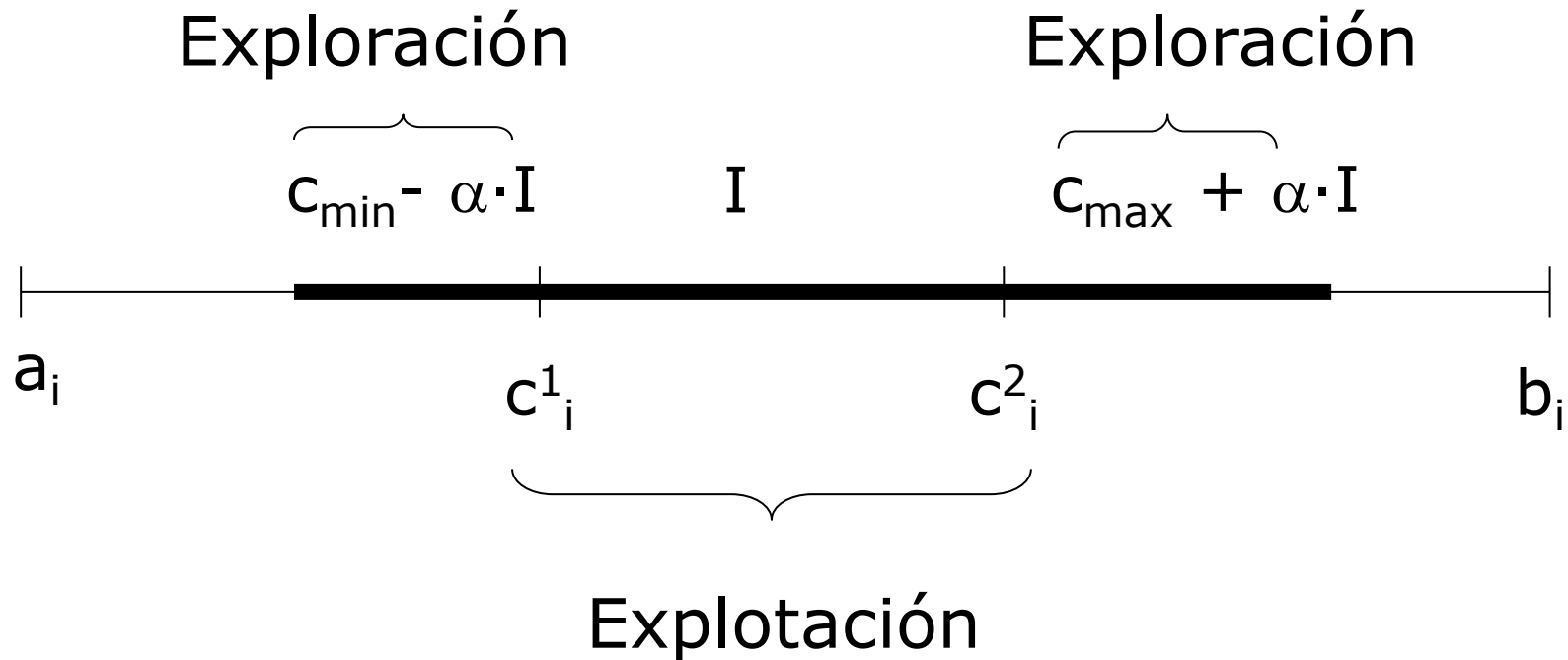
Modelos ampliados: Arithmetical crossover (Michalewicz, 1992),
Max-Min Arithmetic operator (Herrera, Lozano, Verdegay, 1995)



Pioneros y trabajos destacados

Variable-wise recombination: Blend Crossover (BLX- α)

Eshelman L.J., Schaffer J.D. (1993). *Real-Coded Genetic Algorithms and Interval-Schemata*. FOGA 2, 187-202.



- ▶ Distribución de probabilidad uniforme con una cota controlada por α
- ▶ Diversidad en los hijos proporcional a la de los padres
- ▶ La búsqueda es demasiado amplia si los padres están distantes

Pioneros y trabajos destacados

Variable-wise recombination of Parents (RCGA first generation)

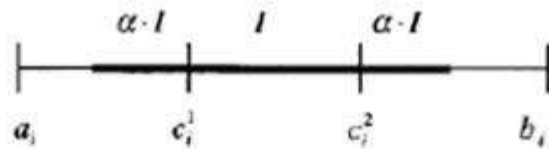
- ▶ Usa distribución de probabilidad para crear descendencia
- ▶ Diferentes implementaciones desde 1991:
 - ▶ Blend crossover (BLX- α), 1993
 - ▶ Simulated binary crossover (SBX- β), 1995
 - ▶ Fuzzy recombination (FR-d), 1995
 - ▶ Fuzzy connectives based operator (FCB), 1994
- ▶ **Característica principal:** La diferencia entre los padres se usa para crear a los hijos
 - ▶ Proporciona una propiedad auto-adaptativa

Análisis experimentales: F. Herrera, M. Lozano, J.L. Verdegay (1998). Tackling real-coded genetic algorithms: operators and tools for the behavioural analysis.

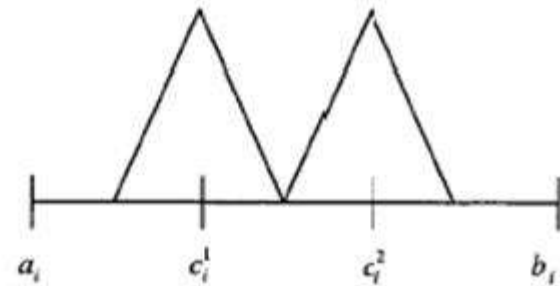
Artificial Intelligence Reviews 12(4): 265-319

Pioneros y trabajos destacados

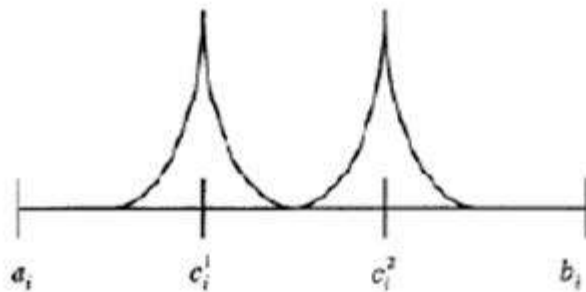
BLX- α (Eshelman et al., 1993)



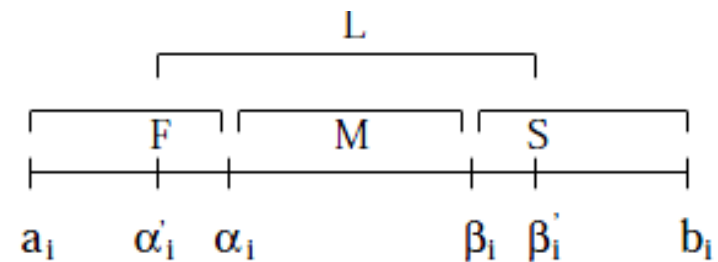
Fuzzy recombination (Voigt et al., 1995)



SBX (Deb et al., 1995)



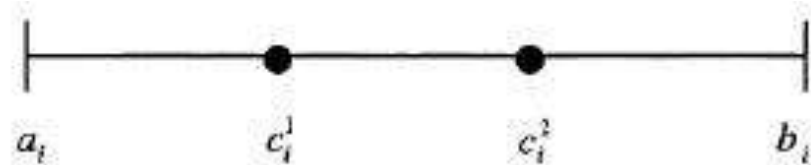
Fuzzy Connectives based Operator (Herrera et al. 1994)



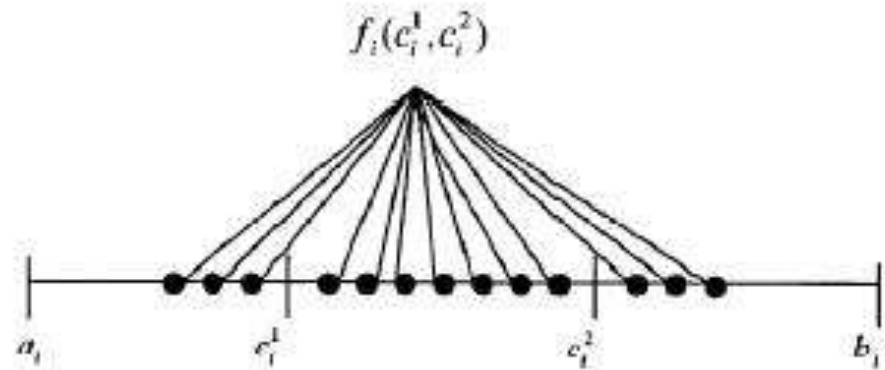
Pioneros y trabajos destacados

Taxonomía de los operadores de cruce

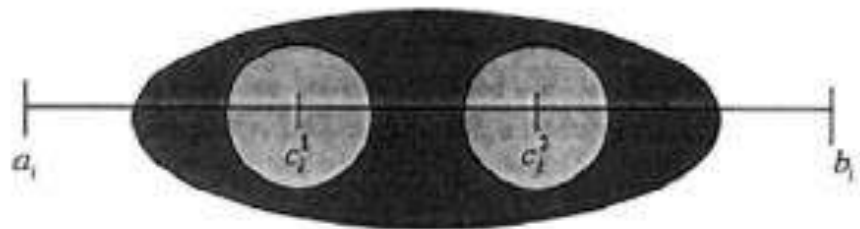
Cruce discreto



Cruce basado en agregación

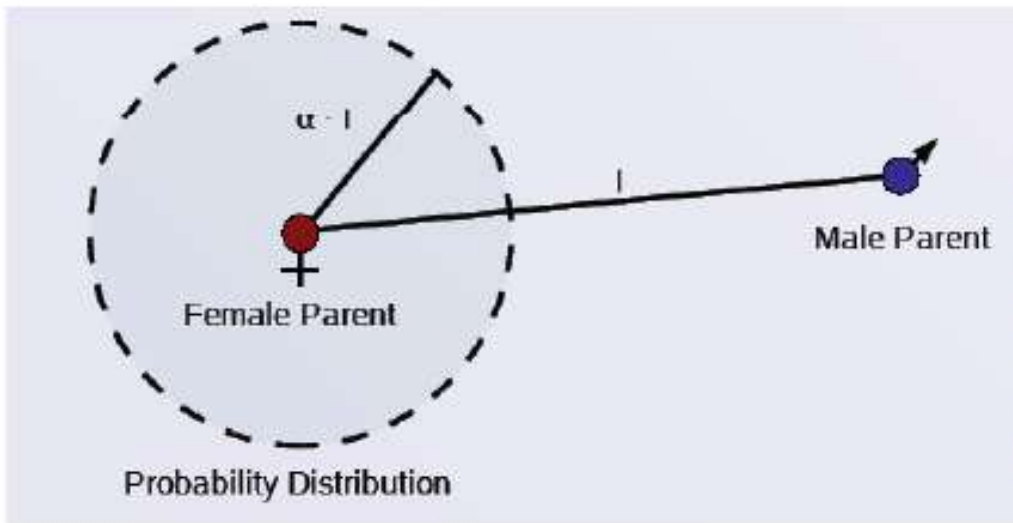


Cruce basado en vecindario



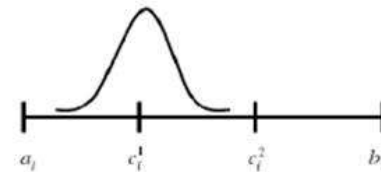
Pioneros y trabajos destacados

Operadores de cruce basados en Parent Center



- FR (Voigt et al, 1995)
- SBX (Deb et al, 1995)
- XLM (Takahashi et al 2001)
- PCX (Deb et al, 2002)
- vSBX (Ballester et al, 2003)
- PNX (Ballester et al, 2004)
- PBX- α (Lozano et al, 2004)

PNX (Ballester et al., 2004)



Comportamiento similar que el de operadores auto-adaptados

Pioneros y trabajos destacados

Operadores de Vector-Wise Recombination

- ▶ Variable-wise recombination no puede capturar interacciones no lineales
- ▶ **Alternativa:** Recombinar padres como vectores (**RCGA second generation**)
 - ▶ Parent-centric recombination (PCX)
 - ▶ Unimodal normally-distributed crossover (UNDX)
 - ▶ Simplex crossover (SPX)
- ▶ La diferencia entre los padres se usa para crear las soluciones descendentes (**algunos modelos en este caso especial**).

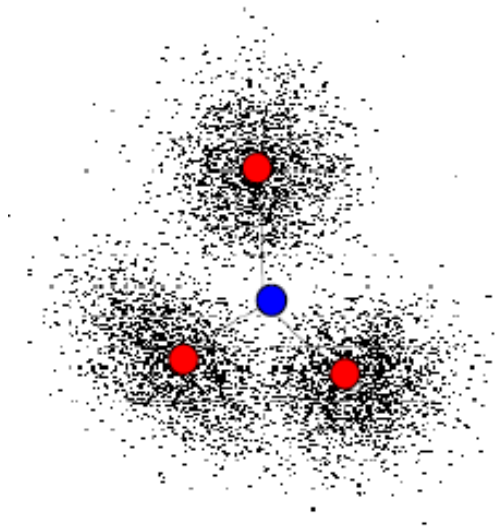


F. Herrera, M. Lozano (Eds.) (2005). Special Issue on Real Coded Genetic Algorithms: Foundations, Models and Operators. Soft Computing 9:4.

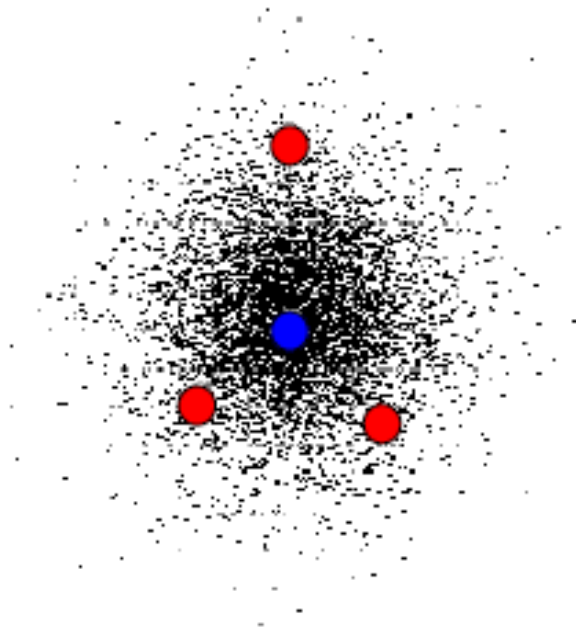
Pioneros y trabajos destacados

Recombinar a los parents como vectores Operadores PCX, UNDX y SPX

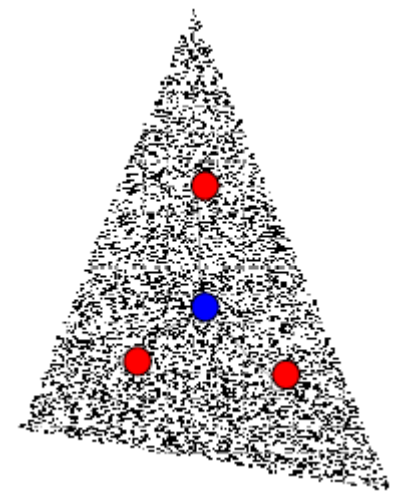
PCX



UNDX



SPX



Pioneros y trabajos destacados

Operadores de Vector-Wise Recombination

- ▶ La recombinación Variable-wise no puede capturar las interacciones no lineales.
- ▶ **Alternativa:** Recombinar a los padres como vectores (2ª generación RCGA)
 - ▶ Recombinación Parent-centric (PCX)
 - ▶ Cruce Unimodal Normalmente Distribuido (UNDX)
 - ▶ Cruce Simple (SPX)
- ▶ La diferencia entre los padres es usada para crear (algunos modelos en este caso especial).
- ▶ **Nuevos algoritmos (2ª generación EAs): DE, PSO, CMA-ES**

METAHEURÍSTICAS

TEMA 3. METAHEURÍSTICAS BASADAS EN POBLACIONES

Parte II:

1. ALGORITMOS EVOLUTIVOS PARA OPTIMIZACIÓN CONTINUA I (Algoritmos genéticos)
2. **ESTRATEGIAS DE EVOLUCIÓN**
3. EVOLUCIÓN DIFERENCIAL
4. TEMA 6. PSO. ALGORITMOS DE NUBES DE PARTÍCULAS
5. ALGORITMOS EVOLUTIVOS PARA OPTIMIZACIÓN CONTINUA (Competiciones y modelos)
6. NUEVOS MODELOS BIOINSPIRADOS PARA OPTIMIZACIÓN DE PARÁMETROS

Pioneros y trabajos destacados

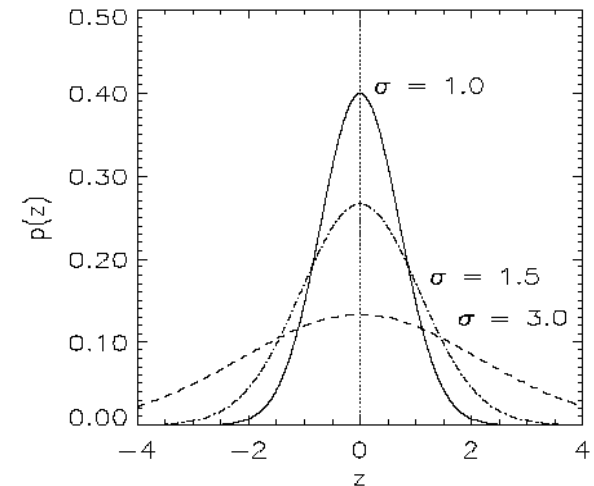
Estrategias de evolución

Rechenberg & Schwefel (1964) estaban preocupados por resolver problemas de optimización de parámetros. Auto-adaptación de los parámetros.

$$\text{Mut: } \mathbf{I} \longrightarrow \mathbf{I} \\ (\mathbf{x}_1, \dots, \mathbf{x}_n)$$

$$\text{Mut}(\mathbf{x}) = \mathbf{x}' = (\mathbf{x}_1 + \mathbf{z}_1, \dots, \mathbf{x}_n + \mathbf{z}_n)$$

$$\mathbf{z}_i \sim N_i(0, \sigma^2)$$



State of the art of the first generation: Schwefel, H.P. *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology Series. Wiley, New York, 1995.

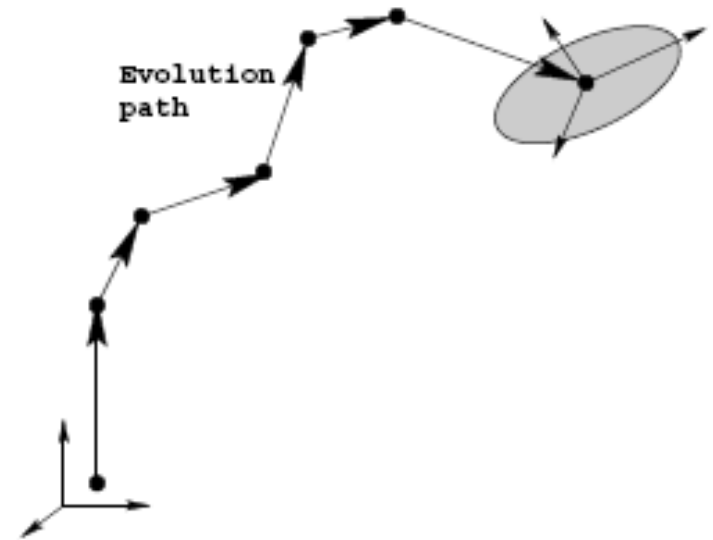
Pioneros y trabajos destacados

State of the art of the ES second generation: CMA-ES

Evolution Strategy with Covariance Matrix Adaptation (Hansen & Ostermeier, 1996)

- ▶ Selection-mutation ES se ejecuta para n iteraciones
- ▶ Se registran los pasos exitosos
- ▶ Son analizados para encontrar direcciones básicas y puntos fuertes no correlacionados
- ▶ Requiere $O(n^3)$ para solucionar un problema de autovalores
- ▶ Rotación invariante

Nikolaus Hansen
www.lri.fr/~hansen/



- Hansen, N. and A. Ostermeier (2001). Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2), pp. 159-195;
- Hansen, N., S.D. Müller and P. Koumoutsakos (2003). Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 11(1), pp. 1-18;

METAHEURÍSTICAS

TEMA 3. METAHEURÍSTICAS BASADAS EN POBLACIONES

Parte II:

1. ALGORITMOS EVOLUTIVOS PARA OPTIMIZACIÓN CONTINUA I (Algoritmos genéticos)
2. ESTRATEGIAS DE EVOLUCIÓN
3. EVOLUCIÓN DIFERENCIAL
4. TEMA 6. PSO. ALGORITMOS DE NUBES DE PARTÍCULAS
5. ALGORITMOS EVOLUTIVOS PARA OPTIMIZACIÓN CONTINUA II (Competiciones y modelos)
6. NUEVOS MODELOS BIOINSPIRADOS PARA OPTIMIZACIÓN DE PARÁMETROS

METAHEURÍSTICAS

TEMA 3. METAHEURÍSTICAS BASADAS EN POBLACIONES

EVOLUCIÓN DIFERENCIAL

1. EVOLUCIÓN DIFERENCIAL
2. VARIANTES DE LA EVOLUCIÓN DIFERENCIA
3. MODELOS AVANZANDOS EN EVOLUCIÓN DIFERENCIAL

1. EVOLUCIÓN DIFERENCIAL

- Es un modelo evolutivo que enfatiza la mutación, utiliza un operador de cruce/recombinación a posteriori de la mutación. Fué propuesto para optimización con parámetros reales.
- Fue propuesta of R. Storm (Univ. Berkeley), 1997

R. Storn, Differential Evolution, A simple and eficiente heuristic strategy for global optimization over continuous spaces. Journal of Global Optimization, 11 (1997) 341-359.



Kenneth V. Price, Rainer M. Storn, and Jouni A. Lampinen
[Differential Evolution: A Practical Approach to Global Optimization \(Natural Computing Series\)](#)
Springer-Verlag, 2005.

1. EVOLUCIÓN DIFERENCIAL

- **Inicialización:**

Una población $P_{x,0}$ de N_p vectores de parámetros D-dimensionales $x_{i,0}=[x_{1,i,0}, \dots, x_{D,i,0}]$, $i=1, \dots, N_p$ se genera aleatoriamente dentro de unos límites inferiores y superiores previos $b_L=[b_{1,L}, \dots, b_{D,L}]$ y $b_U=[b_{1,U}, \dots, b_{D,U}]$.

Ejemplo: el valor inicial (en la generación $g=0$) del j-ésimo parámetro del i-ésimo vector se genera por: $x_{j,i,0}=\text{rand}_j[0, 1] \cdot (b_{j,U}-b_{j,L}) + b_{j,L}$, $j=1, \dots, D$, $i=1, \dots, N_p$.

- **Generación:**

Mutación Diferencial: con respecto a cada vector $x_{i,g}$ en la población actual, llamado vector objetivo, se genera un vector mutado $v_{i,g}$ añadiendo un vector diferencia, escalado y aleatoriamente muestreado, a un vector base aleatoriamente seleccionado de la población actual.

En la generación g-ésima, se genera una población $P_{u,g}$ consistente de N_p vectores D-dimensionales $u_{i,g}=[u_{1,i,g}, \dots, u_{D,i,g}]$ a través de operadores de mutación y recombinación aplicados a la población actual $P_{x,g}$.

1. EVOLUCIÓN DIFERENCIAL

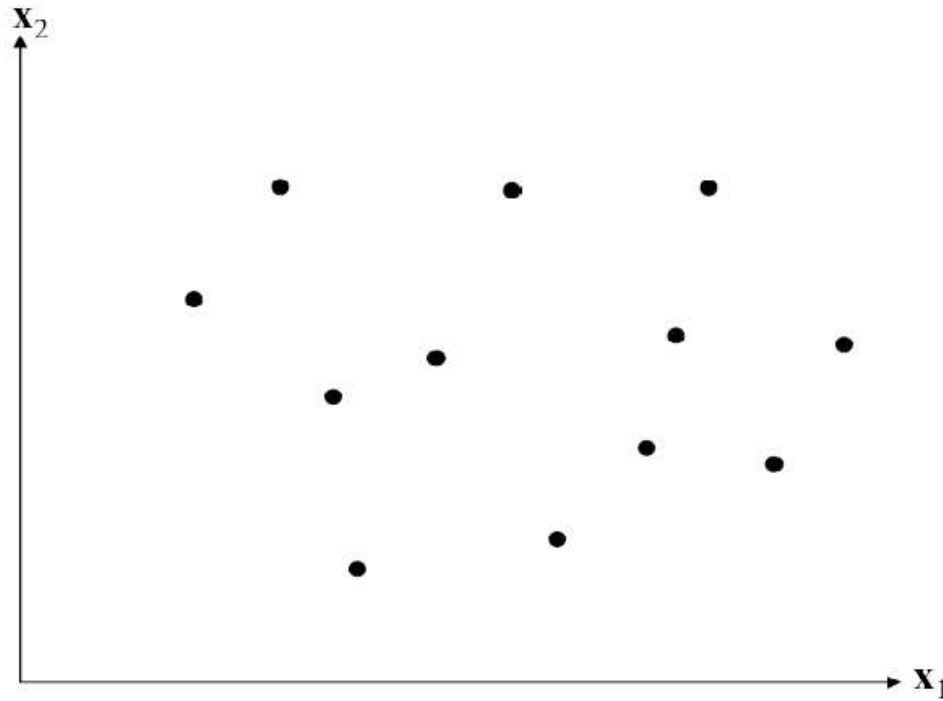


Imagen de la Evolución Diferencial clásica

1. EVOLUCIÓN DIFERENCIAL

Ejemplo: en la generación g -ésima, el i -ésimo vector mutado $v_{i,g}$ con respecto al i -ésimo vector objetivo $x_{i,g}$ en la población actual se genera mediante

$$V_{i,g} = x_{r0,g} + F \cdot (x_{r1,g} - x_{r2,g}), \quad i \neq r0 \neq r1 \neq r2.$$

El factor de escalado de la mutación $F \in [0,1]$

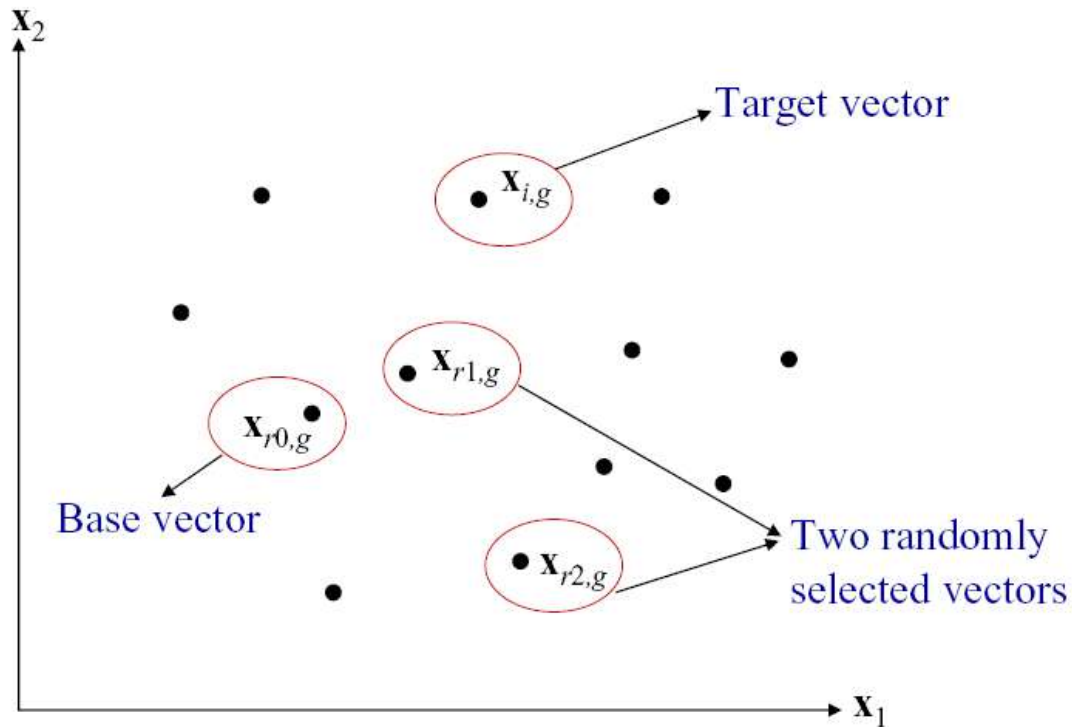


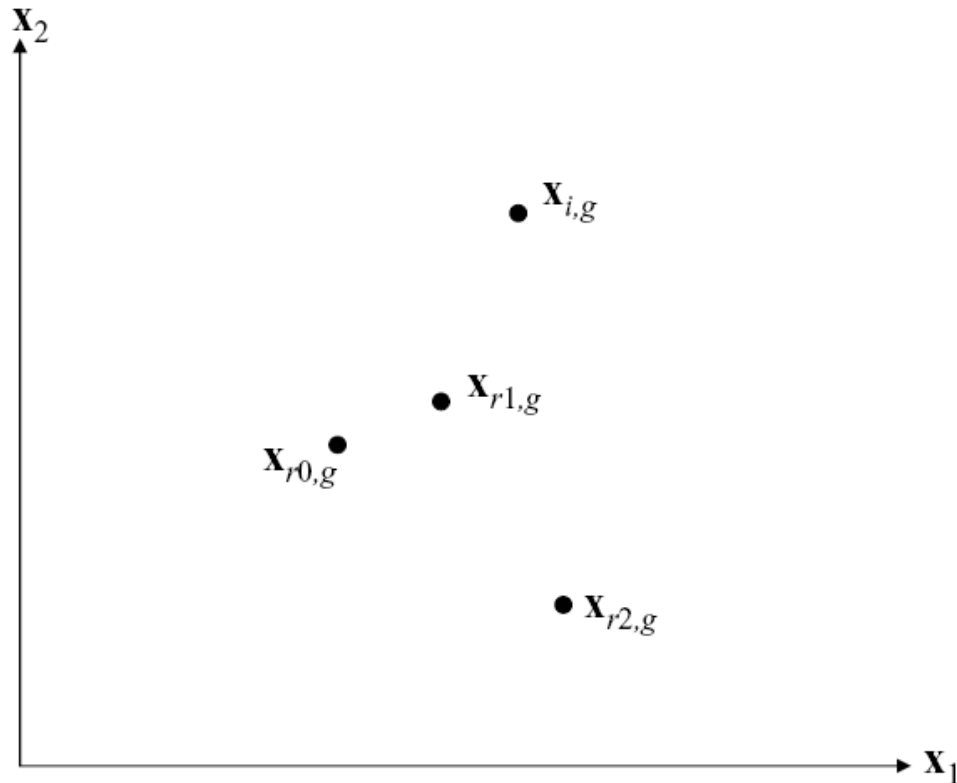
Illustration of classic DE

1. EVOLUCIÓN DIFERENCIAL

Ejemplo: en la generación g -ésima, el i -ésimo vector mutado $v_{i,g}$ con respecto al i -ésimo vector objetivo $x_{i,g}$ en la población actual se genera mediante

$$V_{i,g} = x_{r0,g} + F \cdot (x_{r1,g} - x_{r2,g}), \quad i \neq r0 \neq r1 \neq r2.$$

El factor de escalado de la mutación $F \in [0,1]$

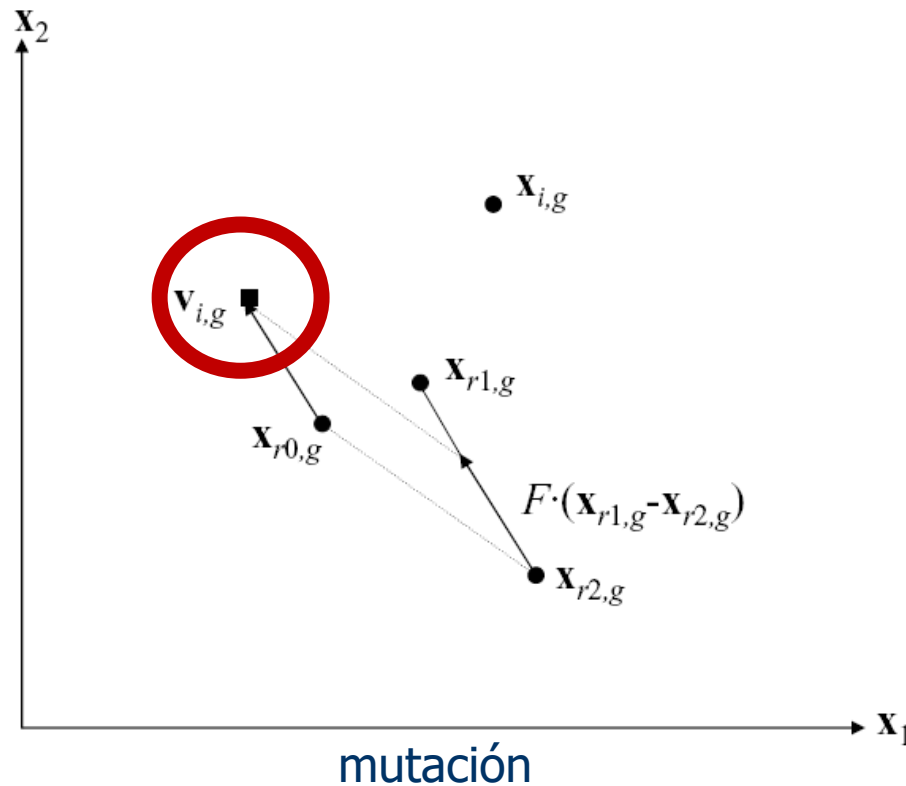


1. EVOLUCIÓN DIFERENCIAL

Ejemplo: en la generación g -ésima, el i -ésimo vector mutado $v_{i,g}$ con respecto al i -ésimo vector objetivo $x_{i,g}$ en la población actual se genera mediante

$$V_{i,g} = x_{r0,g} + F \cdot (x_{r1,g} - x_{r2,g}), \quad i \neq r0 \neq r1 \neq r2.$$

El factor de escalado de la mutación $F \in [0,1]$



1. EVOLUCIÓN DIFERENCIAL

- **Recombinación Discreta:**

Con respecto a cada vector objetivo $x_{i,g}$ en la población actual, un nuevo vector $u_{i,g}$ se genera cruzando el vector objetivo $x_{i,g}$ con el correspondiente vector mutado $v_{i,g}$ bajo un ratio predefinido de cruce $Cr \in [0, 1]$.

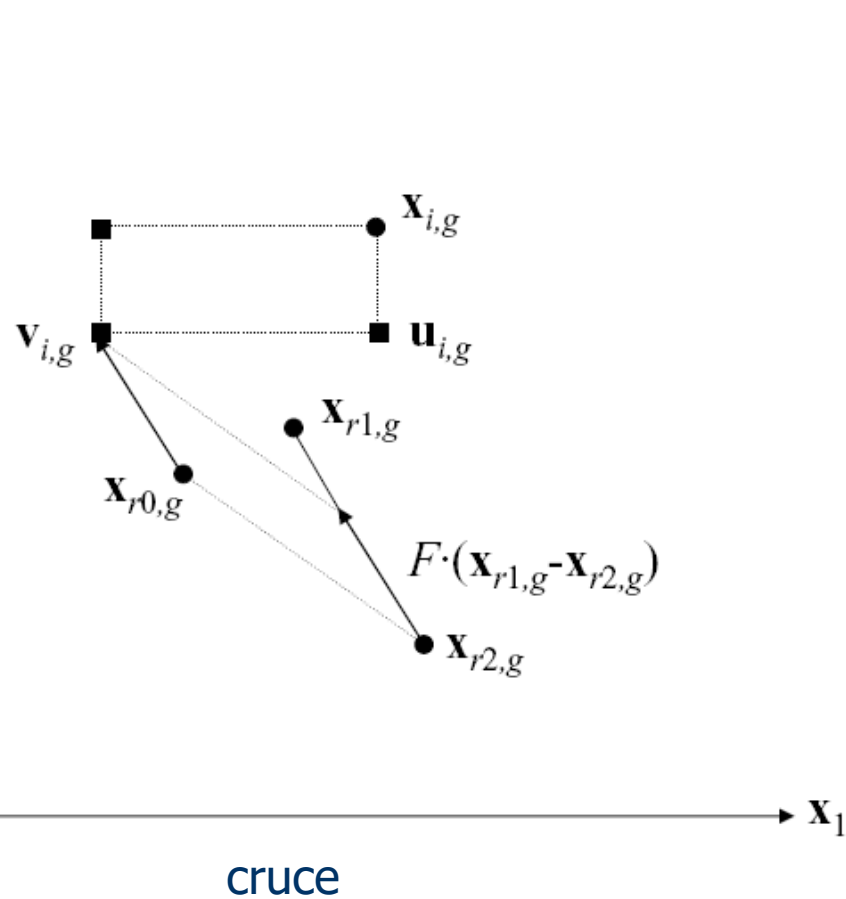
Ejemplo: en la generación g -ésima, el i -ésimo vector $u_{i,g}$ con respecto al i -ésimo vector objetivo $x_{i,g}$ en la población actual se genera mediante:

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if } \text{rand}_j[0,1] \leq Cr \\ x_{j,i,g} & \text{otherwise} \end{cases}$$

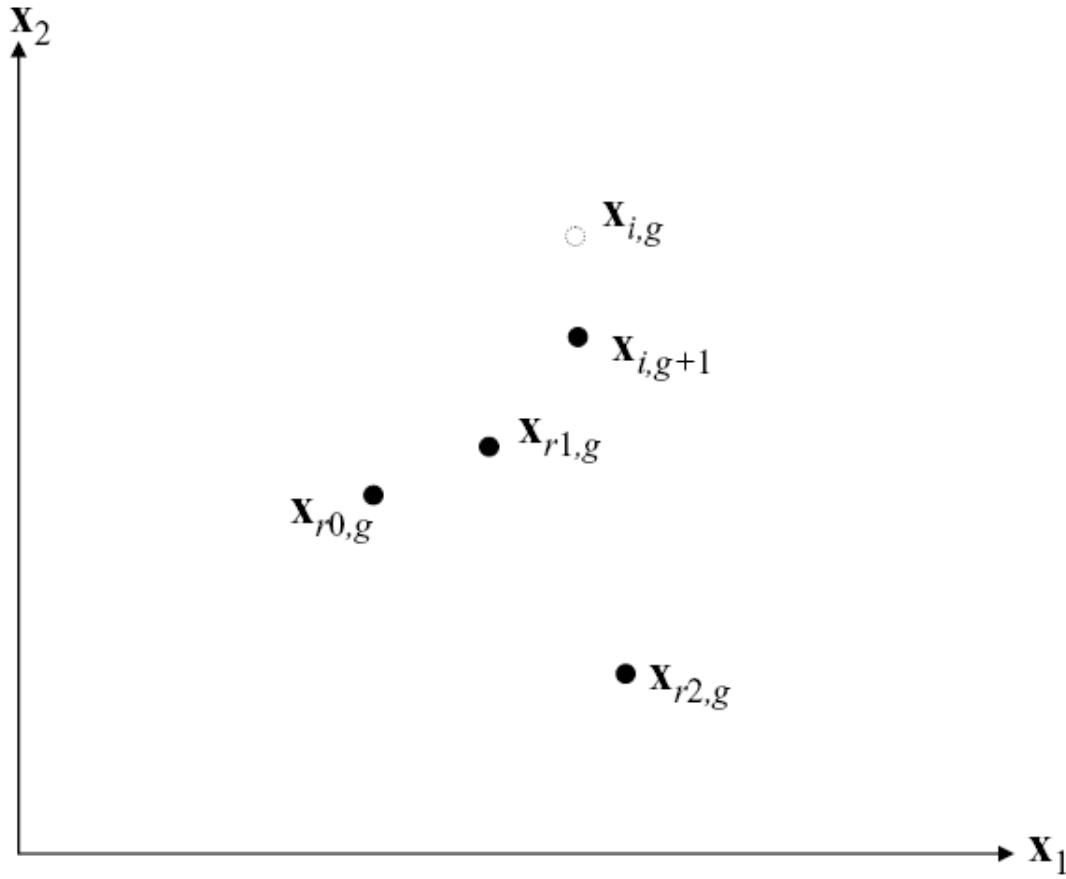
- **Reemplazamiento:**

Si el vector $u_{i,g}$ tiene mejor valor de la función objetivo que su correspondiente vector objetivo $x_{i,g}$, sustituye el vector objetivo en la generación $(g+1)$; si esto no ocurre, el vector objetivo permanece en la generación $(g+1)$.

1. EVOLUCIÓN DIFERENCIAL



1. EVOLUCIÓN DIFERENCIAL



sustitución

1. EVOLUCIÓN DIFERENCIAL

Procedimiento Básico – Evolución Diferencial

```
Procedure DE{
  t = 0;
  Initialize Pop(t);    /* of |Pop(t)| Individuals */
  Evaluate Pop(t);
  While (Not Done)
  {for i = 1 to |Pop(t)| do
    {parent1, parent2, parent3} = Select_3_Parents(Pop(t));
    thisGene = random_int(|Pop(t)|);
    for k = 1 to n do /* n genes per Individual */
      if (random < p) *P is crossover constant in [0,1]*
        Offspringik = parent1ik +  $\Upsilon$ (parent2ik - parent3ik);
      else
        Offspringik = Individualik in Pop(t);
    end /* for k */
    Evaluate(Offspringi);
  end /* for i */
  Pop(t+1) = {j | Offspringj is_better_than Individualj}  $\cup$ 
             {k | Individualk is_better_than Offspringk};
  t = t + 1;}
```

CÓDIGO: <http://www.icsi.berkeley.edu/~storn/code.html>

2. VARIANTES DE LA EVOLUCIÓN DIFERENCIAL

- Mutación diferencial:
 - Vector de una diferencia: $F \cdot (x_{r1} - x_{r2})$
 - Vector de dos diferencias: $F \cdot (x_{r1} - x_{r2}) + F \cdot (x_{r3} - x_{r4})$
 - Factor de escalado de mutación F
 - Rol crucial: balance exploración y explotación.

$$\text{DE/rand/1:} \quad \mathbf{V}_{i,G} = \mathbf{X}_{r1,G} + F \cdot (\mathbf{X}_{r2,G} - \mathbf{X}_{r3,G})$$

$$\text{DE/best/1:} \quad \mathbf{V}_{i,G} = \mathbf{X}_{best,G} + F \cdot (\mathbf{X}_{r1,G} - \mathbf{X}_{r2,G})$$

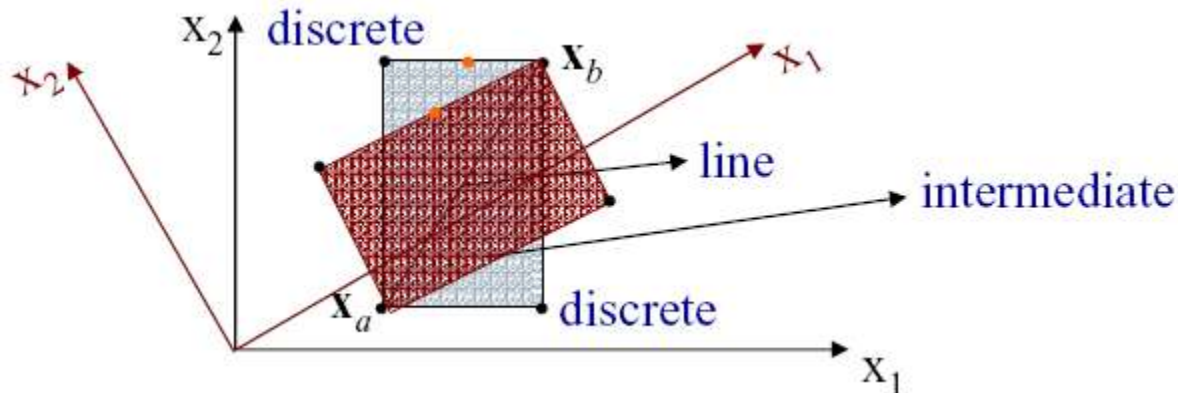
$$\text{DE/current-to-best/1:} \quad \mathbf{V}_{i,G} = \mathbf{X}_{i,G} + F \cdot (\mathbf{X}_{best,G} - \mathbf{X}_{i,G}) + F \cdot (\mathbf{X}_{r1,G} - \mathbf{X}_{r2,G})$$

$$\text{DE/rand/2:} \quad \mathbf{V}_{i,G} = \mathbf{X}_{r1,G} + F \cdot (\mathbf{X}_{r2,G} - \mathbf{X}_{r3,G} + \mathbf{X}_{r4,G} - \mathbf{X}_{r5,G})$$

$$\text{DE/best/2:} \quad \mathbf{V}_{i,G} = \mathbf{X}_{best,G} + F \cdot (\mathbf{X}_{r1,G} - \mathbf{X}_{r2,G} + \mathbf{X}_{r3,G} - \mathbf{X}_{r4,G})$$

2. VARIANTES DE LA EVOLUCIÓN DIFERENCIAL

- Recombinación
 - Recombinación discreta (cruce) (variante rotacional)
 - Un punto y multipunto
 - Exponencial
 - Binomial (uniforme)
 - Recombinación aritmética
 - Recombinación lineal (invariante rotacional)
 - Recombinación intermedia (variante rotacional)
 - Recombinación intermedia extendida (variante rotacional)



2. VARIANTES DE LA EVOLUCIÓN DIFERENCIAL

Ratio de cruce $CR \in [0, 1]$

Descomponible (CR pequeño) y funciones no descomponibles (CR grande)

Guía práctica:

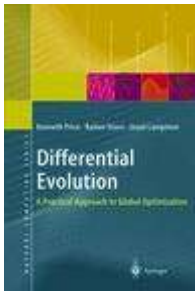
$N_p \in [5D, 10D]$; elección inicial de $F = 0.5$ y $CR = 0.1 / 0.9$;

Incrementar N_p y / o F si encontramos convergencia prematura.

3. ALGUNOS TRABAJOS EN EVOLUCIÓN DIFERENCIAL



R. Storn and K. V. Price, “Differential evolution-A simple and Efficient Heuristic for Global Optimization over Continuous Spaces,” *Journal of Global Optimization*, 11:341-359,1997.



K. V. Price, R. Storn, J. Lampinen, *Differential Evolution - A Practical Approach to Global Optimization*, Springer, Berlin, 2005.

3. MODELOS AVANZADOS EN EVOLUCIÓN DIFERENCIAL

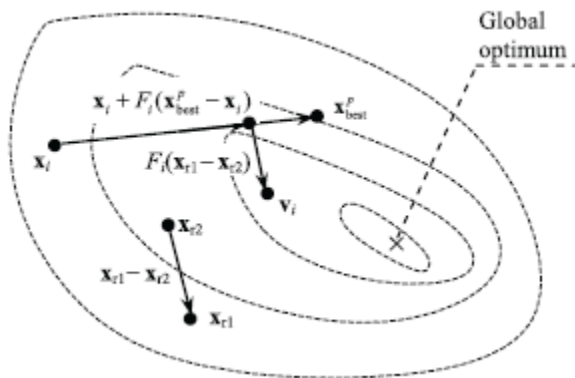
SaDE: Self adaptive Differential Evolution

- Para alcanzar los mejores resultados en un problema, es frecuente usar una búsqueda prueba-error para dar valor a esos parámetros: CR, F, NP. Se requiere mucho tiempo.
- SaDE se propone para ajustar tanto soluciones como valores de parámetros.
- Se consideran varias estrategias y se elige aquella que funcionó mejor en generaciones previas.
 - De alta convergencia: DE/rand-to-best/2/bin.
 - De lenta convergencia: DE/rand/1/bin.
 - De dos diferencias, para ofrecer mejores perturbaciones: DE/rand/2/bin
 - Estrategia de rotación invariante: DE/current-to-rand/1.
- La probabilidad de elección de cada estrategia se adapta mediante una memoria de fallos y aciertos.
- SaDE adapta los parámetros CR y F. Inicialización: $F \sim N(0.5, 0.3)$, $CR \sim N(0.5, 0.1)$

3. ALGUNOS TRABAJOS CLÁSICOS EN EVOLUCIÓN DIFERENCIAL

JADE: Adaptive Differential Evolution

- Utiliza un esquema DE/current-to-pbest y autoadapta los parámetros F y CR.



$$\mathbf{v}_{i,g} = \mathbf{x}_{i,g} + F_i \cdot (\mathbf{x}_{best,g}^p - \mathbf{x}_{i,g}) + F_i \cdot (\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g})$$

- Donde se escoge aleatoriamente uno de los 100p% mejores.
- \mathbf{x}_{r2} puede escogerse entre la población y un archivo opcional, que mantiene soluciones no seleccionadas aleatoriamente en generaciones anteriores.
- Autoadaptación de parámetros similar a SaDE.

COMENTARIOS FINALES

Los algoritmos de Evolución Diferencial son uno de los campos más activos en el desarrollo de algoritmos evolutivos para la optimización de parámetros (optimización continua).

<http://www.sciencedirect.com/science/article/pii/S2210650216000146>



Swarm and Evolutionary Computation

Volume 27, April 2016, Pages 1–30



Survey Paper

Recent advances in differential evolution – An updated survey

Swagatam Das^a,  , Sankha Subhra Mullick^a,  , P.N. Suganthan^b,  

 [Show more](#)

COMENTARIOS FINALES

Los algoritmos de Evolución Diferencial son uno de los campos más activos en el desarrollo de algoritmos evolutivos para la optimización de parámetros (optimización continua).

Ryoji Tanabe, Ph.D.

<https://ryojitanabe.github.io/>

Source code

L-SHADE was a first ranked method at the CEC2014 Competition on Real-Parameter Single Objective Optimization

SHADE

- An improved version of JADE [Zhang 09]
- Uses a different parameter adaptation mechanism based on the success-history based adaptation
- L-SHADE = SHADE + Linear population reduction method

[1] Ryoji Tanabe and Alex Fukunaga: Improving the Search Performance of SHADE Using Linear Population Size Reduction, Proc. IEEE Congress on Evolutionary Computation (CEC-2014), Beijing, July, 2014 pp. 1658-1665.

COMENTARIOS FINALES

Los algoritmos de Evolución Diferencial son uno de los campos más activos en el desarrollo de algoritmos evolutivos para la optimización de parámetros (optimización continua).

<http://www.sciencedirect.com/science/article/pii/S0020025514009438>



Information Sciences

Volume 316, 20 September 2015, Pages 517–549

Nature-Inspired Algorithms for Large Scale Global Optimization



A comprehensive comparison of large scale global optimizers

Antonio LaTorre^a, , , Santiago Muelas^b, , José-María Peña^b, 

*State of the art for large scale optimization: **MOS VARIANTS**: A MOS-based Dynamic **Memetic Differential Evolution** Algorithm for Continuous Optimization*

METAHEURÍSTICAS

TEMA 3. METAHEURÍSTICAS BASADAS EN POBLACIONES

Parte II:

1. ALGORITMOS EVOLUTIVOS PARA OPTIMIZACIÓN CONTINUA I (Algoritmos genéticos)
2. EVOLUCIÓN DIFERENCIAL
3. ESTRATEGIAS DE EVOLUCIÓN
4. **TEMA 6. PSO. ALGORITMOS DE NUBES DE PARTÍCULAS**
5. ALGORITMOS EVOLUTIVOS PARA OPTIMIZACIÓN CONTINUA (Competiciones y modelos)
6. NUEVOS MODELOS BIOINSPIRADOS PARA OPTIMIZACIÓN DE PARÁMETROS

METAHEURÍSTICAS

OPTIMIZACIÓN BASADA EN NUBES DE PARTÍCULAS (PARTICLE SWARM)

- 1. INTRODUCCIÓN Y RÁPIDO RESUMEN**
- 2. FUNCIONAMIENTO DEL ALGORITMO PSO**
- 3. ASPECTOS AVANZADOS**

Kennedy, J., Eberhart, R.C. Swarm Intelligence. Morgan Kauffmann, 2001.

1. INTRODUCCIÓN Y RÁPIDO RESUMEN

- La “Particle Swarm Optimization” (PSO) es una metaheurística poblacional inspirada en el comportamiento social del vuelo de las bandadas de aves y el movimiento de los bancos de peces.
- La población se compone de varias partículas (**nube de partículas = particle swarm**) que se mueven (“*vuelan*”) por el espacio de búsqueda durante la ejecución del algoritmo.
- Este movimiento de cada partícula p depende de:
 - Su mejor posición desde que comenzó el algoritmo ($pBest$),
 - la mejor posición de las partículas **de su entorno** ($lBest$) o de **toda la nube** ($gBest$) desde que comenzó el algoritmo.

En cada iteración, se cambia aleatoriamente la velocidad de p para acercarla a las posiciones $pBest$ y $lBest/gBest$.

1. INTRODUCCIÓN Y RÁPIDO RESUMEN (2)

- Desarrollo: USA, en 1995.
- Primeros autores: Russ C. Eberhart y James Kennedy
Kennedy, J. and Eberhart, R. (1995). "Particle Swarm Optimization", Proc. 1995 IEEE Intl. Conf. on Neural Networks, pp. 1942-1948, IEEE Press.
- Aplicación típica:
 - Optimización continua (optimización de parámetros reales, numérica).
- Características atribuidas:
 - Asume un intercambio de información (**interacciones sociales**) entre los agentes de búsqueda.
 - **Idea básica**: guardar información del mejor propio y global.
 - Implementación muy sencilla, pocos parámetros.
 - Convergencia rápida a buenas soluciones.

2. FUNCIONAMIENTO DEL ALGORITMO PSO

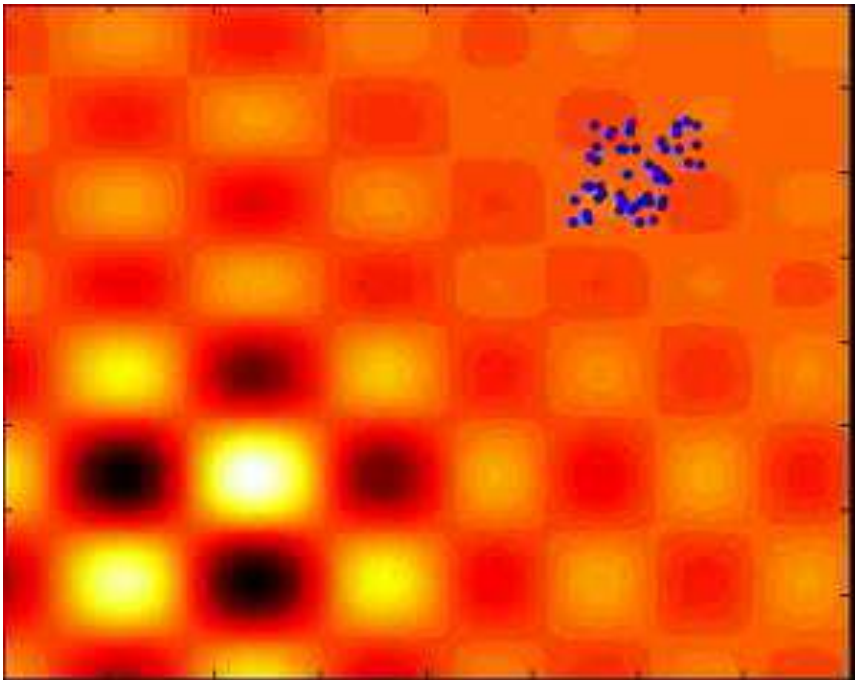
- **FUNCIONAMIENTO BÁSICO**
- **ANATOMÍA DE UNA PARTÍCULA**
- **INICIALIZACIÓN DE LA NUBE DE PARTÍCULAS**
- **MOVIMIENTO DE LAS PARTÍCULAS**
- **PSEUDOCÓDIGOS**
- **VALORES DE LOS PARÁMETROS**
- **TOPOLOGÍAS DE LA NUBE DE PARTÍCULAS**

Funcionamiento Básico

- PSO simula el comportamiento de las bandadas de aves.
- Supongamos que una de estas bandadas busca comida en un área y que solamente hay una pieza de comida en dicha área.
- Los pájaros no saben donde está la comida pero sí conocen su distancia a la misma.
- La estrategia más eficaz para hallar la comida es seguir al ave que se encuentre más cerca de ella.

Funcionamiento Básico (2)

- PSO emula este escenario para resolver problemas de optimización. Cada solución (**partícula**) es un "*ave*" en el espacio de búsqueda que está siempre en continuo movimiento y que nunca muere.



Funcionamiento Básico (2)

■ La nube de partículas es un **sistema multiagente**. Las partículas son agentes simples que se mueven por el espacio de búsqueda y que guardan (y posiblemente comunican) la mejor solución que han encontrado.

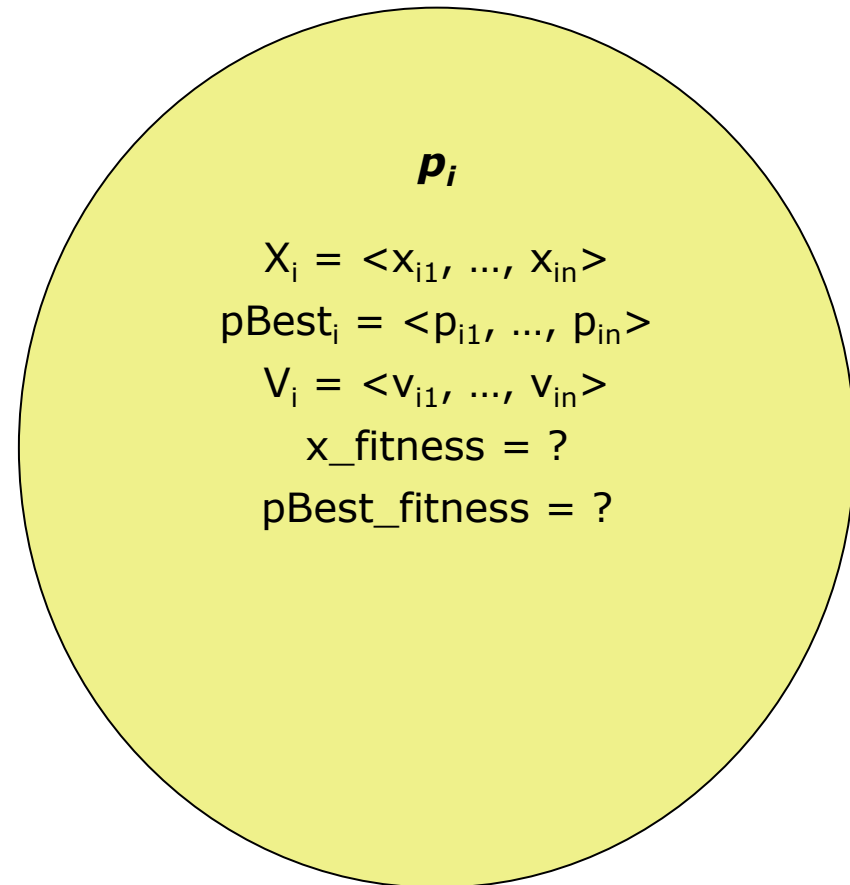


■ Cada partícula tiene un **fitness**, una **posición** y un **vector velocidad** que dirige su "vuelo". El movimiento de las partículas por el espacio está guiado por las partículas óptimas en el momento actual.

Anatomía de una Partícula

Una partícula está compuesta por:

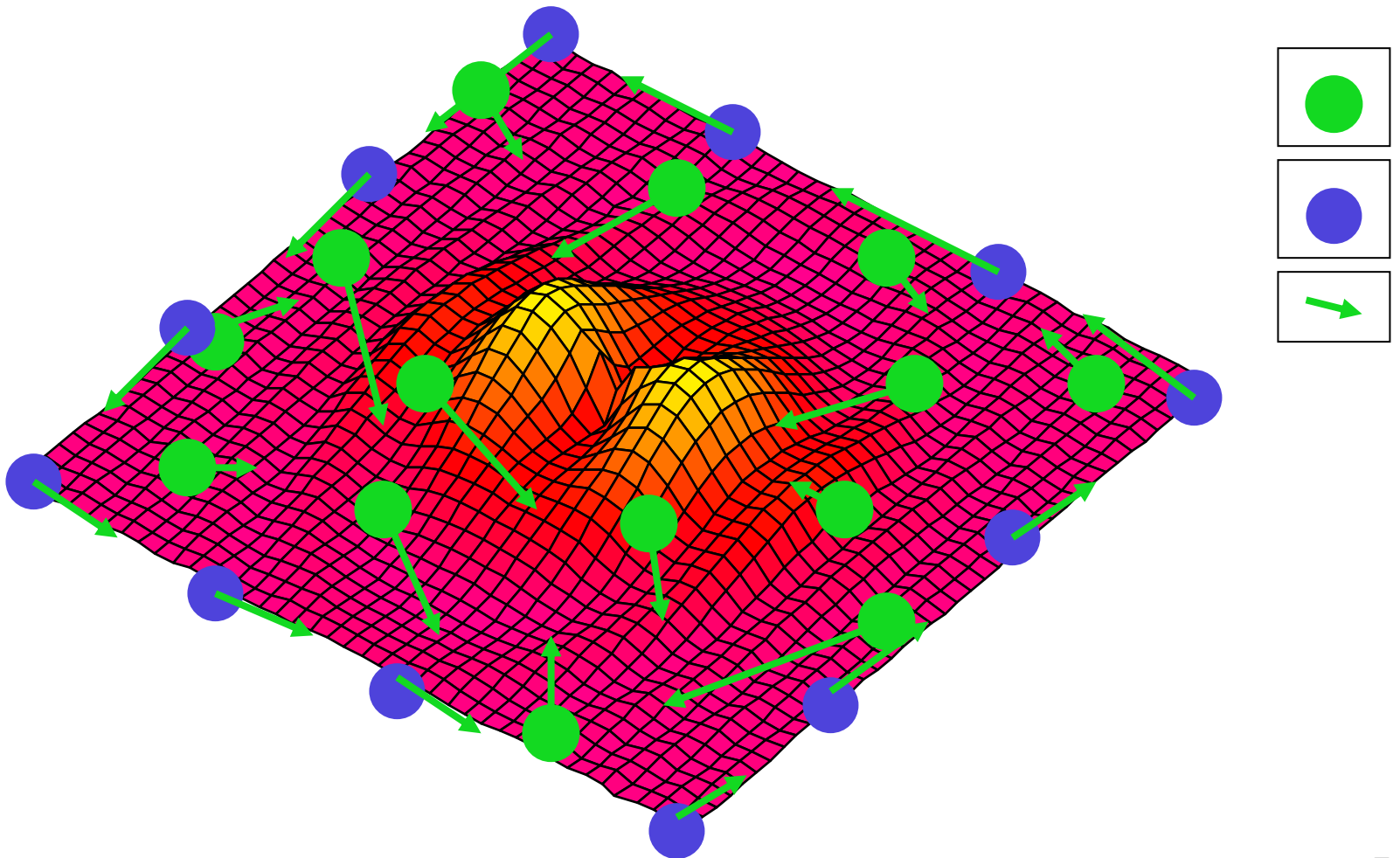
- Tres vectores:
 - El **vector X** almacena la posición actual (localización) de la partícula en el espacio de búsqueda,
 - El **vector pBest** almacena la localización de la mejor solución encontrada por la partícula hasta el momento, y
 - El **vector V** almacena el gradiente (dirección) según el cuál se moverá la partícula.
- Dos valores de fitness:
 - El **x_fitness** almacena el fitness de la solución actual (vector X), y
 - El **p_fitness** almacena el fitness de la mejor solución local (vector *pBest*).



Inicialización de la Nube de Partículas

- La nube se inicializa generando las posiciones y las velocidades iniciales de las partículas.
- Las posiciones se pueden generar aleatoriamente en el espacio de búsqueda, de forma regular, o con una combinación de ambas.
- Las velocidades se generan aleatoriamente, con cada componente en el intervalo $[-V_{\max}, V_{\max}]$.
No es conveniente fijarlas a cero, no se obtienen buenos resultados.
 V_{\max} será la velocidad máxima que pueda tomar una partícula en cada movimiento.

Inicialización de la Nube de Partículas (2)



Movimiento de las Partículas

¿Cómo se mueve una partícula de una posición del espacio de búsqueda a otra?

- Se hace simplemente añadiendo el vector velocidad V_i al vector posición X_i para obtener un nuevo vector posición:

$$X_i \leftarrow X_i + V_i$$

- Una vez calculada la nueva posición de la partícula, se evalúa ésta. Si el nuevo fitness es mejor que el que la partícula tenía hasta ahora, $pBest_fitness$, entonces:

$$pBest_i \leftarrow X_i \quad ; \quad pBest_fitness \leftarrow x_fitness.$$

Movimiento de las Partículas (2)

- De este modo, el primer paso es ajustar el vector velocidad, para después sumárselo al vector posición.
- Las fórmulas empleadas son las siguientes:

$$v_{id} = v_{id} + \underbrace{\varphi_1 \cdot \text{rnd}() \cdot (pBest_{id} - x_{id})}_{\text{COGNITIVO}} + \underbrace{\varphi_2 \cdot \text{rnd}() \cdot (g_{id} - x_{id})}_{\text{SOCIAL}}$$
$$x_{id} = x_{id} + v_{id}$$

donde:

- p_i es la partícula en cuestión, $pBest_{id}$ es la mejor solución encontrada por la partícula.
- φ_1, φ_2 son ratios de aprendizaje (pesos) que controlan los componentes **cognitivo** y **social**,
- g representa el índice de la partícula con el mejor $pBest_fitness$ del entorno de p_i ($iBest$) o de toda la nube ($gBest$),
- los **rnd()** son números aleatorios generados en $[0,1]$, y
- d es la d -ésima dimensión del vector.

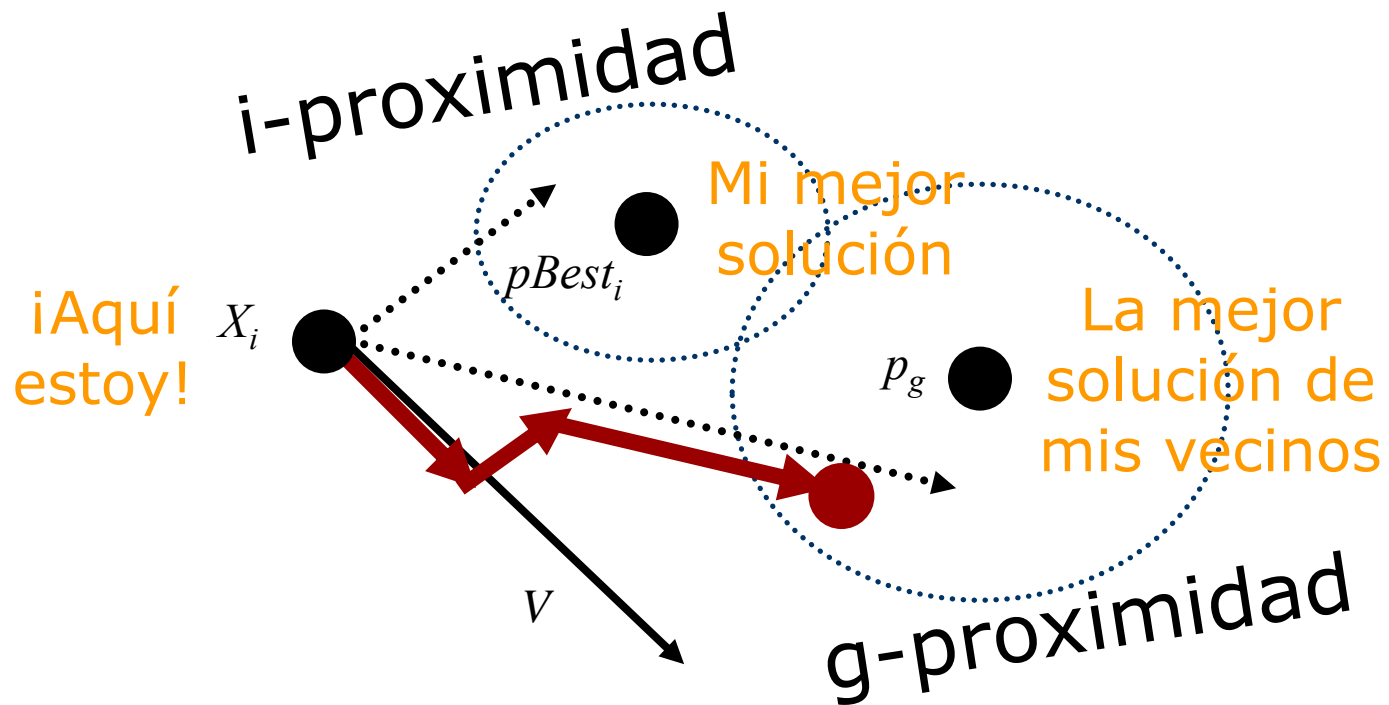
Movimiento de las Partículas (3)

TIPOS DE ALGORITMOS DE PSO:

- Kennedy identifica cuatro tipos de algoritmos de PSO en función de los valores de φ_1 y φ_2 :
 - Modelo completo: $\varphi_1, \varphi_2 > 0$.
 - Sólo Cognitivo: $\varphi_1 > 0$ y $\varphi_2 = 0$.
 - Sólo Social: $\varphi_1 = 0$ y $\varphi_2 > 0$.
 - Sólo Social exclusivo: $\varphi_1 = 0$, $\varphi_2 > 0$ y $g \neq i$ (la partícula en sí no puede ser la mejor de su entorno).

Movimiento de las Partículas (4)

REPRESENTACIÓN GRÁFICA:



Pseudocódigo PSO Local

$t = 0;$

Para $i=1$ hasta Número_partículas
 inicializar X_i y V_i ;

Mientras (no se cumpla la condición de parada) hacer
 $t \leftarrow t + 1$

 Para $i=1$ hasta Número_partículas
 evaluar X_i ;

 Si $F(X_i)$ es mejor que $F(pBest_i)$ entonces
 $pBest_i \leftarrow X_i$; $F(pBest_i) \leftarrow F(X_i)$

 Para $i=1$ hasta Número_partículas

 Escoger $lBest_i$, la partícula con mejor fitness del entorno de X_i

 Calcular V_i , la velocidad de X_i , de acuerdo a $pBest_i$ y $lBest_i$

 Calcular la nueva posición X_i , de acuerdo a X_i y V_i

Devolver la mejor solución encontrada

Pseudocódigo PSO Global

$t = 0;$

Para $i=1$ hasta Número_partículas
inicializar X_i y V_i ;

Mientras (no se cumpla la condición de parada) hacer
 $t \leftarrow t + 1$

Para $i=1$ hasta Número_partículas

evaluar X_i ;

Si $F(X_i)$ es mejor que $F(pBest)$ entonces

$pBest_i \leftarrow X_i; F(pBest_i) \leftarrow F(X_i)$

Si $F(pBest)$ es mejor que $F(gBest)$ entonces

$gBest \leftarrow pBest_i; F(gBest_i) \leftarrow F(pBest_i)$

Para $i=1$ hasta Número_partículas

Calcular V_i , la velocidad de X_i , de acuerdo a $pBest_i$ y **$gBest_i$** ;

Calcular la nueva posición X_i , de acuerdo a X_i y V_i

Devolver la mejor solución encontrada

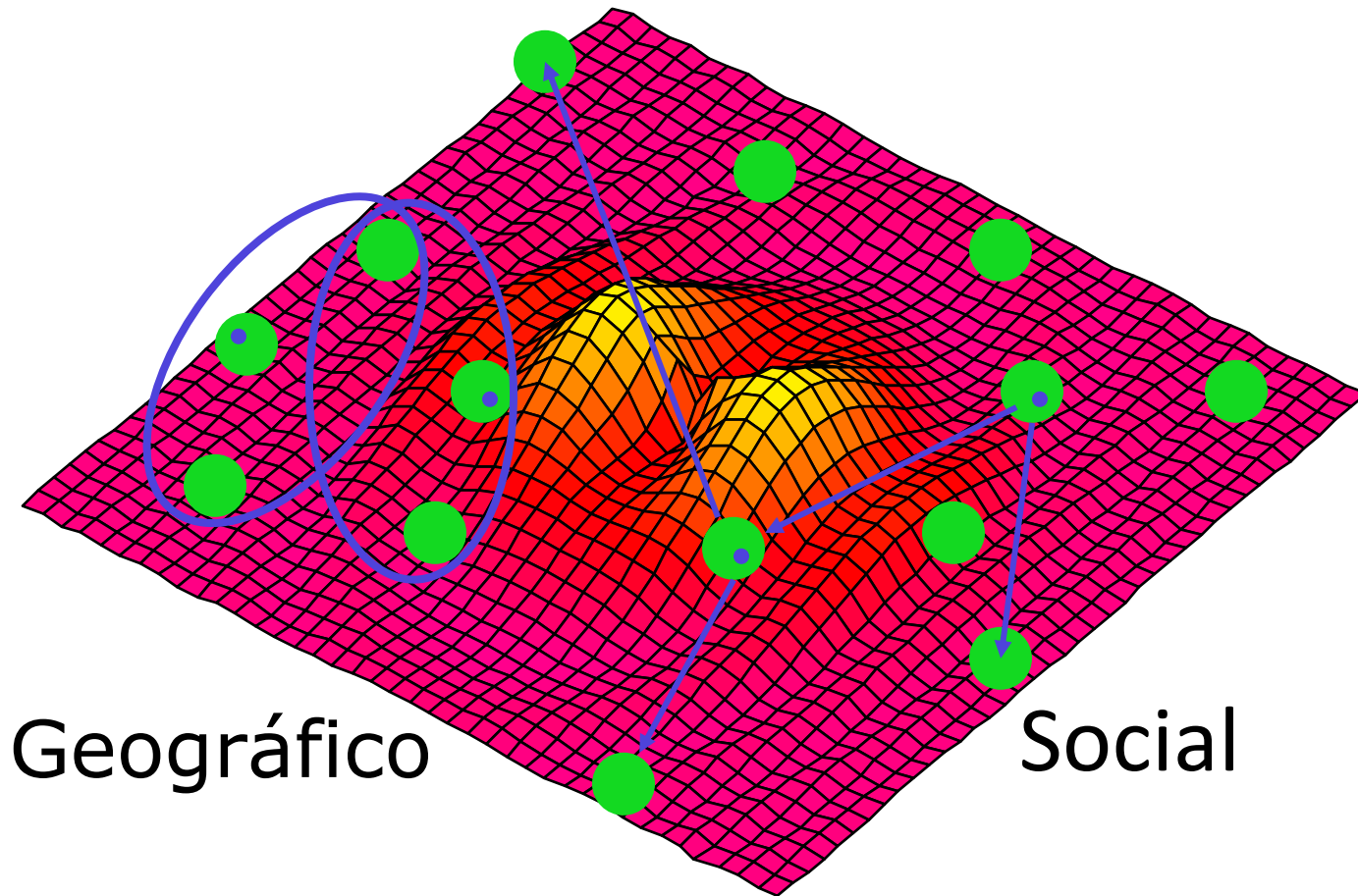
Valores de los Parámetros

- **Tamaño de la nube:** Entre 20 y 40 partículas (problemas simples, 10; problemas muy complejos, 100-200).
- **Velocidad máxima:** V_{\max} se suele definir a partir del intervalo de cada variable.
- **Ratios de aprendizaje:** Habitualmente, $\varphi_1 = \varphi_2 = 2$.
- **PSO Global vs. PSO Local:** La versión global converge más rápido pero cae más fácilmente en óptimos locales y viceversa.

Topologías de la Nube de Partículas

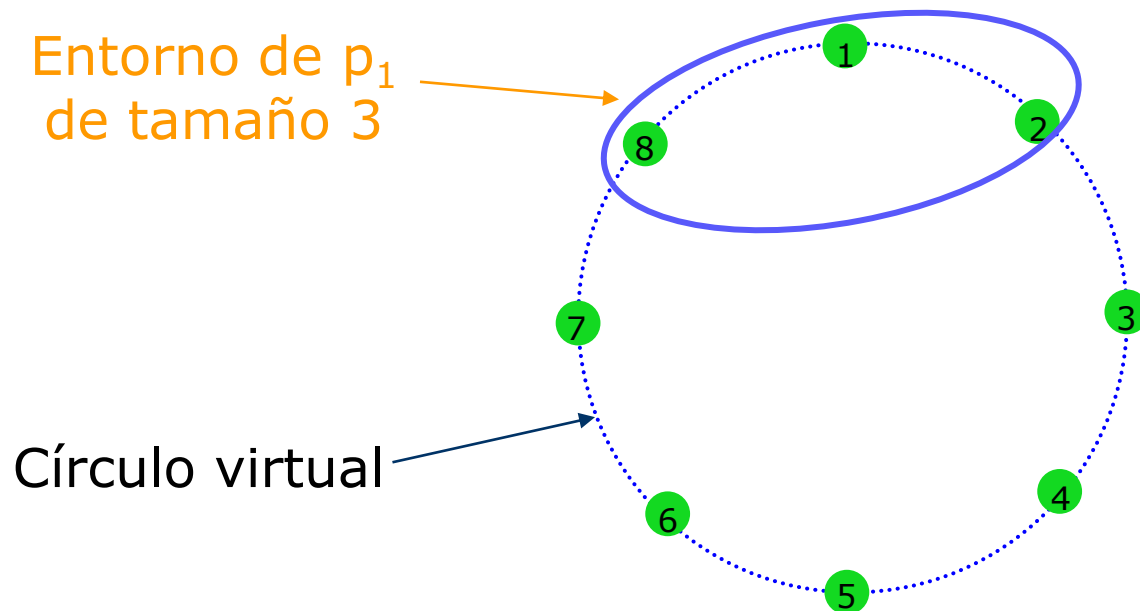
- Las topologías definen el entorno de cada partícula individual. La propia partícula siempre pertenece a su entorno.
- Los entornos pueden ser de dos tipos:
 - Geográficos: se calcula la distancia de la partícula actual al resto y se toman las más cercanas para componer su entorno.
 - Sociales: se define a priori una lista de vecinas para partícula, independientemente de su posición en el espacio.
- Los entornos sociales son los más empleados.
- Una vez decidido el entorno, es necesario definir su tamaño. El algoritmo no es muy sensible a este parámetro (3 o 5 son valores habituales con buen comportamiento).
- Cuando el tamaño es toda la nube de partículas, el entorno es a la vez geográfico y social, y tenemos la PSO global.

Topologías de la Nube de Partículas (2)



Topologías de la Nube de Partículas (3)

- La topología social más empleada es la de anillo, en la que se considera un vecindario circular.
- Se numera cada partícula, se construye un círculo virtual con estos números y se define el entorno de una partícula con sus vecinas en el círculo:



3. ASPECTOS AVANZADOS

- CONTROL DE LA VELOCIDAD DE LAS PARTÍCULAS
- TAMAÑO DE LA NUBE DE PARTÍCULAS
- INFLUENCIA DEL TIPO DE ENTORNO
- ACTUALIZACIÓN DE LAS PARTÍCULAS
- ELECCIÓN DE VALORES ADAPTATIVOS PARA ϕ_1 Y ϕ_2
- ALGUNAS VARIANTES

METAHEURÍSTICAS

TEMA 3. METAHEURÍSTICAS BASADAS EN POBLACIONES

Parte II:

1. ALGORITMOS EVOLUTIVOS PARA OPTIMIZACIÓN CONTINUA I (Algoritmos genéticos)
2. EVOLUCIÓN DIFERENCIAL
3. ESTRATEGIAS DE EVOLUCIÓN
4. TEMA 6. PSO. ALGORITMOS DE NUBES DE PARTÍCULAS
5. ALGORITMOS EVOLUTIVOS PARA OPTIMIZACIÓN CONTINUA II (Competiciones y modelos)
6. NUEVOS MODELOS BIOINSPIRADOS PARA OPTIMIZACIÓN DE PARÁMETROS

Milestone: CEC'2005 Real Parameter

Optimization Session and Benchmark

Special Session on Real-Parameter Optimization.

2005 IEEE CEC, Edinburgh, UK, Sept 2-5. 2005.

Organizers: K. Deb and P.N. Suganthan.

Unimodal Functions

Success Performance Indices

Multimodal Functions

Solved in at least one run

Multimodal Functions

Never solved

El estudio se hizo con dimensiones **$D = 10$, $D = 30$, $D=50$** .

El número máximo de evaluaciones de aptitud es **$10,000 \cdot D$** .

Cada vuelta para cuando el número máximo de evaluaciones se alcanza

P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger and S. Tiwari, "[Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization](#)", *Technical Report*, Nanyang Technological University, Singapore, May 2005 AND KanGAL Report #2005005, IIT Kanpur, India.

Milestone: CEC'2005 Real Parameter

Optimization Session and Benchmark

Special Session on Real-Parameter Optimization.

2005 IEEE CEC, Edinburgh, UK, Sept 2-5, 2005.

Organizers: K. Deb and P.N. Suganthan.

Unimodal Functions

Success Performance Indices

6 funciones

- 1 Sphere
- 2 Schwefel 1.2
- 3 Ellipsoid Condition 10^6
- 4 Schwefel 1.2 with Noise
- 5 Schwefel 2.6 on Bounds
- 6 Rosenbrock

Multimodal Functions

Solved in at least one run

6 funciones

- 7 Griewank out Bounds
- 9 Rastrigin Separable
- 10 Rastrigin Rotated
- 11 Weierstrass
- 12 Schwefel 2.13
- 15 Hybrid Separable

Multimodal Functions

Never solved

13 funciones

- 8 Ackley Condition 10^2
- 13 Expanded 6&7
- 14 Expanded Schaffer F6
- 16 Hybrid Rotated
- 17 Hybrid with Noise
- 18 Hybrid F18
- 19 Hybrid Narrow
- 20 Hybrid on Bounds
- 21 Hybrid F21
- 22 Hybrid High Condition
- 23 Hybrid Non-Continuous
- 24 Hybrid F24
- 25 Hybrid out

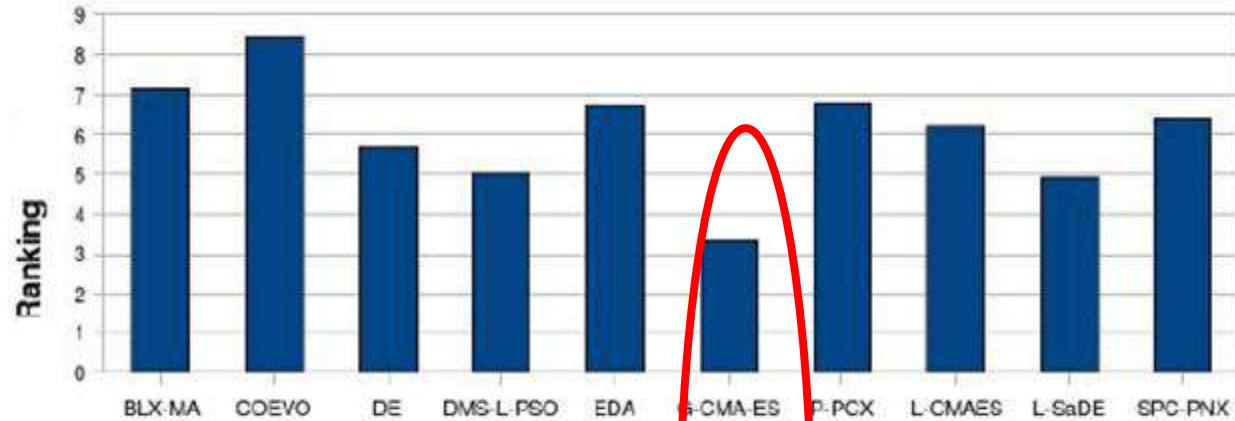
Milestone: CEC'2005 Real Parameter

Optimization Session and Benchmark

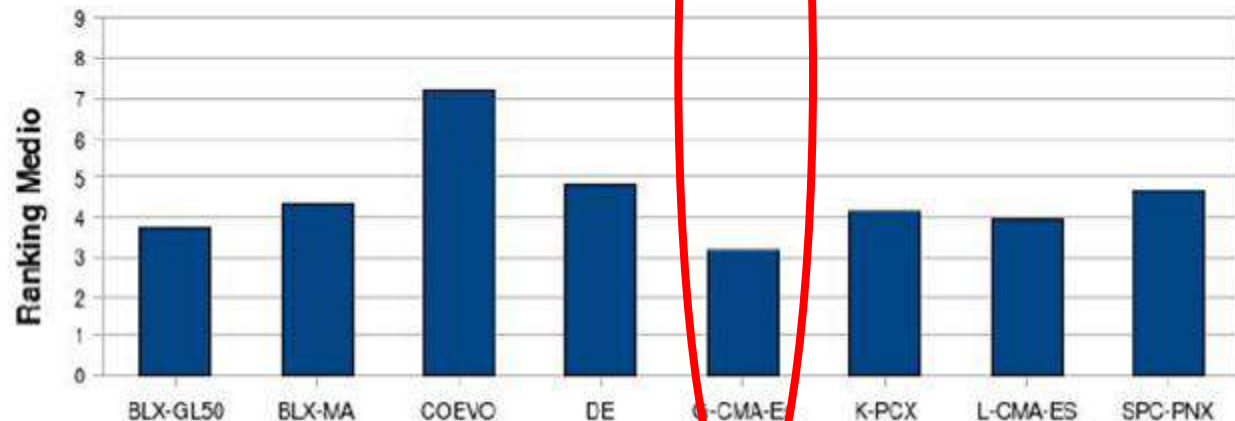
- Algorithms involved in the comparison: **(11 algoritmos)**
 - **BLX-GL50** (Garcia-Martinez & Lozano, 2005): Hybrid Real-Coded Genetic Algorithms with Female and Male Differentiation
 - **BLX-MA** (Molina *et al.*, 2005): Adaptive Local Search Parameters for Real-Coded Memetic Algorithms
 - **CoEVO** (Posik, 2005): Mutation Step Co-evolution
 - **DE** (Ronkkonen *et al.*,2005):Differential Evolution
 - **DMS-L-PSO**: Dynamic Multi-Swarm Particle Swarm Optimizer with Local Search
 - **EDA** (Yuan & Gallagher, 2005): Estimation of Distribution Algorithm
 - **G-CMA-ES** (Auger & Hansen, 2005): A restart Covariance Matrix Adaptation Evolution Strategy with increasing population size
 - **K-PCX** (Sinha *et al.*, 2005): A Population-based, Steady-State real-parameter optimization algorithm with parent-centric recombination operator, a polynomial mutation operator and a niched -selection operation.
 - **L-CMA-ES** (Auger & Hansen, 2005): A restart local search Covariance Matrix Adaptation Evolution Strategy
 - **L-SaDE** (Qin & Suganthan, 2005): Self-adaptive Differential Evolution algorithm with Local Search
 - **SPC-PNX** (Ballester *et al.*,2005): A steady-state real-parameter GA with PNX crossover operator

Milestone: CEC'2005 Real Parameter Optimization Session and Benchmark

$D = 10$



$D = 30$



Milestone: CEC'2005 Real Parameter Optimization Session and Benchmark

G-CMA-ES vs.	R^+	R^-	p -value
BLX-GL50	289.5	35.5	0.001
BLX-MA	295.5	29.5	0.001
CoEVO	301.0	24.0	0.000
DE	262.5	62.5	0.009
DMS-L-PSO	199.0	126.0	0.357
EDA	284.5	40.5	0.001
K-PCX	269.0	56.0	0.004
L-CMA-ES	273.0	52.0	0.003
L-SaDE	209.0	116.0	0.259
SPC-PNX	305.5	19.5	0.000

G-CMAES versus el resto de algoritmos. $D = 10$

P-valor obtenido mediante la aproximación normal

S. García, [D. Molina](#), [M. Lozano](#), [F. Herrera](#), A Study on the Use of Non-Parametric Tests for Analyzing the Evolutionary Algorithms' Behaviour: A Case Study on the CEC'2005 Special Session on Real Parameter Optimization. *Journal of Heuristics*, 15 (2009) 617-644. [doi: 10.1007/s10732-008-9080-4](https://doi.org/10.1007/s10732-008-9080-4).

Milestone: CEC'2005 Real Parameter

Optimization Session and Benchmark

Dos algoritmos con buen ranking y comportamiento estadístico similar:

AMALGAM – SO: Vrugt, J.A.; Robinson, B.A.; Hyman, J.M.; , "Self-Adaptive Multimethod Search for Global Optimization in Real-Parameter Spaces," Evolutionary Computation, IEEE Transactions on , vol.13, no.2, pp.243-259, April 2009

<http://math.lanl.gov/~vrugt/software/>

AMALGAM - SO: A Multi ALgorithm Genetically Adaptive Method for Single Objective Optimization. Este método combina simultáneamente las ventajas del Covariance Matrix Adaptation (**CMA**) evolution strategy, Genetic Algorithm (**GA**) and Particle Swarm Optimizer (**PSO**) para la evolución de la población e implementa una estrategia de aprendizaje auto-adaptativa para sintonizar automáticamente el número de descendientes con los que a estos tres algoritmos se les permite contribuir durante cada generación.

Milestone: CEC'2005 Real Parameter

Optimization Session and Benchmark

Dos algoritmos con buen ranking y comportamiento estadístico similar:

MA-CMA-Chains: [D. Molina](#), [M. Lozano](#), [C. García-Martínez](#), [F. Herrera](#), Memetic Algorithms for Continuous Optimization Based on Local Search Chains. *Evolutionary Computation*, 18(1), 2010, 27–63.

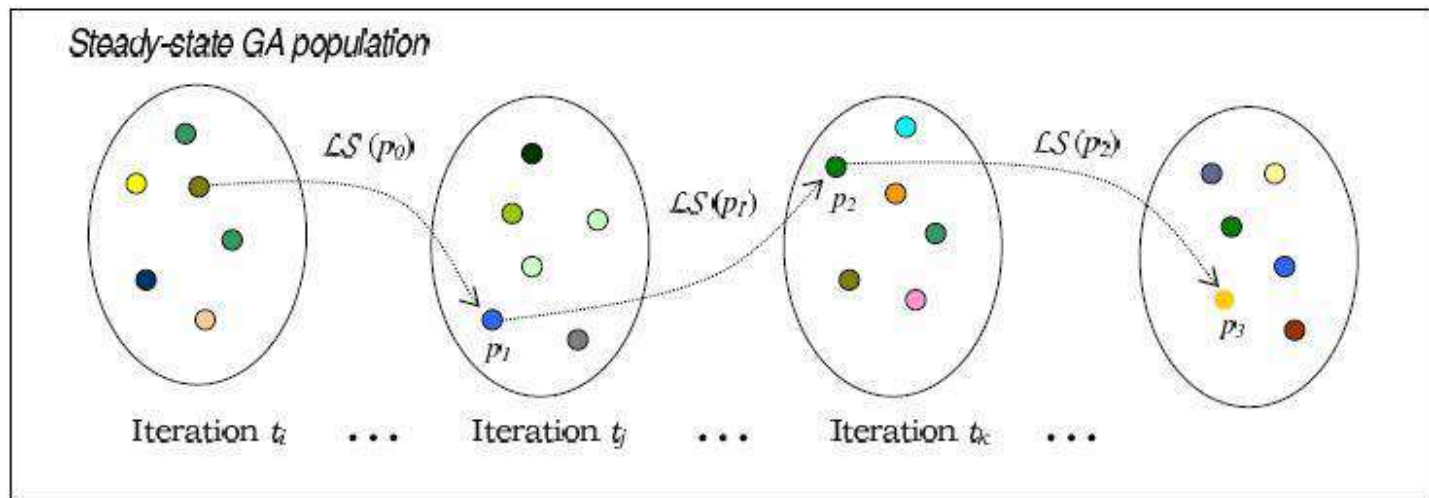


Figure 3: Example of LS chain. p_{i+1} is the final parameter value reached by the LS algorithm when it started with a value of p_i . p_0 is the default value for the strategy parameter

MA-CMA-Chains: Adaptación de búsqueda local

Large Scale Global Optimization



**Special Issue of Soft Computing:
Scalability of Evolutionary Algorithms and other Metaheuristics
for Large Scale Continuous Optimization Problems**
Volume 15, Number 11, 2011

<http://sci2s.ugr.es/EAMHCO/#LSCOP-special-issue-SOCO>

6. Material complementario: SOCO Special Issue on Large Scale Continuous Optimization Problems

- A set of 19 scalable function optimization problems were provided:
 - **6 Funcionts: F1-F6 of the CEC'2008 test suite.** A detailed description may be found in: K. Tang, X. Yao, P. N. Suganthan, C. MacNish, Y. P. Chen, C. M. Chen, and Z. Yang. [Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization](#). Technical Report, Nature Inspired Computation and Applications Laboratory, USTC, China, 2007. ([Source code](#)).
 - **5 Shifted Functions: Schwefel's Problem 2.22 (F7), Schwefel's Problem 1.2 (F8), Extended f10 (F9), Bohachevsky (F10), and Schaffer (F11).** ([Description](#)) ([Source code](#)).
 - **8 Hybrid Composition Functions (F12-F19*):** They are non-separable functions built by combining two functions belonging to the set of functions F1-F11 ([Description](#)) ([Source code](#)).

El estudio fue hecho con dimensiones $D = 50$, $D = 100$, $D=200$, $D=500$, y $D = 1,000$. El número máximo de evaluaciones de aptitud es $5,000 \cdot D$.

Cada vuelta para cuando el número máximo de evaluaciones es alcanzado.

Large Scale Global Optimization



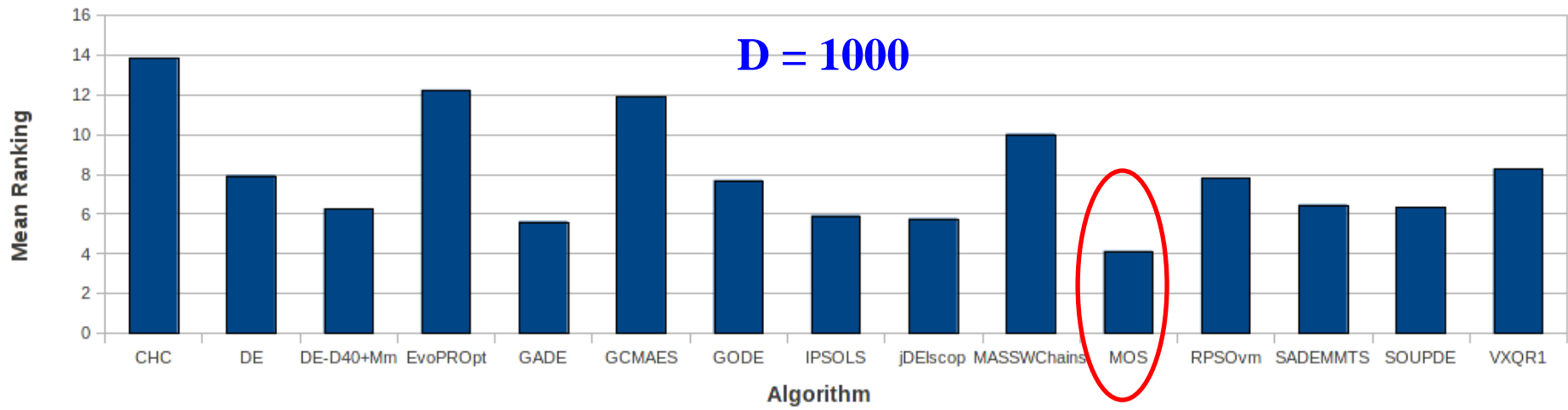
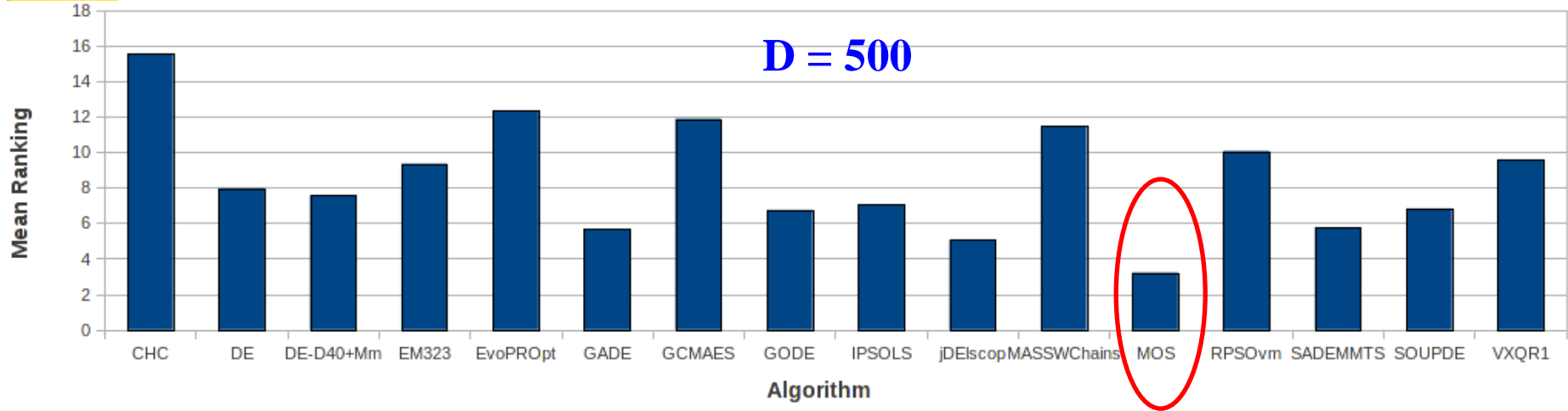
Special Issue of Soft Computing: Scalability of Evolutionary Algorithms and other Metaheuristics for Large Scale Continuous Optimization Problems [Volume 15](#), Number 11, 2011 ([7 DE approaches](#))

- P01 - SOUPDE* Shuffle Or Update **Parallel Differential Evolution** for Large Scale Optimization
- P02 - DE-D⁴⁰+M^m* Role Differentiation and Malleable Mating for **Differential Evolution**: An Analysis on Large Scale Optimisation
- P03 -GODE* Enhanced Opposition-Based **Differential Evolution** for Solving High-Dimensional Continuous Optimization Problems
- P04 - GaDE* Scalability of Generalized Adaptive **Differential Evolution** for Large-Scale Continuous Optimization
- P05 - jDElscop* Self-adaptive **Differential Evolution** Algorithm using Population Size Reduction and Three Strategies
- P06 - SaDE-MMTS* Self-adaptive **Differential Evolution with Multi-trajectory Search** for Large Scale Optimization
- P07 - MOS* A MOS-based Dynamic **Memetic Differential Evolution** Algorithm for Continuous Optimization A Scalability Test (**best results**)
- P08 - MA-SSW-Chains* **Memetic Algorithms Based on Local Search Chains** for Large Scale Continuous Optimisation Problems: MA-SSW-Chains
- P09 - RPSO-vm* **Restart Particle Swarm Optimization** with Velocity Modulation: A Scalability Test
- P10 - Tuned IPSOLS* An Incremental **Particle Swarm** for Large-Scale Optimization Problems: An Example of Tuning-in-the-loop (Re)Design of Optimization Algorithms
- P11 -multi-scale PSO* Multi-Scale **Particle Swarm Optimization** Algorithm
- P12 - EvoPROpt* **Path Relinking** for Large Scale Global Optimization
- P13 - EM323* EM323 : A **Line Search based algorithm** for solving high-dimensional continuous non-linear optimization problems
- P14 - VXQR* VXQR: Derivative-free unconstrained optimization based on **QR factorizations**

Large Scale Global Optimization



Special Issue of Soft Computing: Scalability of Evolutionary Algorithms and other Metaheuristics for Large Scale Continuous Optimization Problems [Volume 15](#), Number 11, 2011



A MOS-based Dynamic Memetic Differential Evolution Algorithm for Continuous Optimization A Scalability Test. *A. LaTorre, S. Muelas, J.M. Peña*. *Soft Computing*, 15, pages: 2187-2199, 2011.

Large Scale Global Optimization



Special Issue of Soft Computing: Scalability of Evolutionary Algorithms and other Metaheuristics for Large Scale Continuous Optimization Problems [Volume 15](#), Number 11, 2011

The algorithm with best values is MOS, in the following Wilcoxon's test we compare this one with the other algorithms,

Algorithm	MOS value	Other value	Critical value p-value 5% error	Sig. differences?
CHC	189,5	0,5	46	Yes
DE	172	18	46	Yes
DE-D40+Mm	157	33	46	Yes
EM323	176	14	46	Yes
EvoPROpt	189,5	0,5	46	Yes
GADE	138	52	46	No
G-CMA-ES	166,5	23,5	46	Yes
GODE	167,5	22,5	46	Yes
IPSOLS	109	81	46	No
JDElscop	143,5	46,5	46	Yes
MASSWChains	182,5	7,5	46	Yes
RPSOvm	176	14	46	Yes
SADEMMTS	132,5	57,5	46	Yes
SOUPDE	157	33	46	Yes
VXQR1	163,5	26,5	46	Yes

D = 500

Large Scale Global Optimization



Special Issue of Soft Computing: Scalability of Evolutionary Algorithms and other Metaheuristics for Large Scale Continuous Optimization Problems [Volume 15](#), Number 11, 2011

The algorithm with best values is MOS, in the following Wilcoxon's test we compare this one with the other algorithms,

D = 1000

Algorithm	MOS value	Other value	Critical value p-value 5% error	Sig. differences?
CHC	189,5	0,5	46	Yes
DE	176	14	46	Yes
DE-D40+Mm	157	33	46	Yes
EvoPROpt	190	0	46	Yes
GADE	138	52	46	No
G-CMA-ES	170,5	19,5	46	Yes
GODE	159	31	46	Yes
IPSOLS	95	95	46	No
JDElscop	153	37	46	Yes
MASSWChains	163,5	26,5	46	Yes
RPSOvm	178	18	46	Yes
SADEMMTS	136,5	53,5	46	No
SOUPDE	167,5	22,5	46	Yes
VXQR1	160,5	29,5	46	Yes

Tendencias actuales: Competiciones **y propuestas**

**COMPETICIONES DE OPTIMIZACIÓN CONTINUA
A LO LARGO DE VARIOS AÑOS**

**COMPETICIONES INCLUYENDO DIFERENTES CONJUNTOS
DE
FUNCIONES DE OPTIMIZACIÓN**

**DIFERENTES MODELOS DE OPTIMIZACIÓN AVANZADOS
QUE HAN IDO MEJORADO ALGUNAS DE LAS FAMILIAS DE
ALGORITMOS**

ECLOSIÓN DE PROPUESTAS DE OPTIMIZACIÓN CONTINUA

Tendencias actuales: Competiciones

Benchmarks for Evaluation of Evolutionary Algorithms

We organized several competitions on benchmarking evolutionary algorithms. Recently, we also developed **several composition functions** to evaluate evolutionary algorithms. The objective of this work is explained in our Swarm Intelligence Symposium 2005 paper and also in the CEC Invited Session / Competition pages listed below.

J. J. Liang, P. N. Suganthan and K. Deb, "[Novel Composition Test Functions for Numerical Global Optimization](#)", *IEEE Swarm Intelligence Symposium*, pp. 68-75, June 2005. [Matlab codes of composition functions](#).

[CEC'05 Special Session / Competition](#) on Evolutionary Real Parameter single objective optimization

[CEC'06 Special Session / Competition](#) on Evolutionary Constrained Real Parameter single objective optimization

[CEC'07 Special Session / Competition](#) on Performance Assessment of real-parameter MOEAs

[CEC'08 Special Session / Competition](#) on large scale single objective global optimization with bound constraints

[CEC'09 Special Session / Competition](#) on Dynamic Optimization (**Primarily composition functions were used**)

[CEC09 Special Session / Competition](#) on Performance Assessment of real-parameter MOEAs

[CEC10 Special Session / Competition](#) on large-scale single objective global optimization with bound constraints

Papers Submitted to CEC-2010 (The last 2 were not published by the CEC 2010)

1. D. Molina, M. Lozano, and F. Herrera, "MA-SW-Chains: Memetic Algorithm Based on Local Search Chains for Large Scale Continuous Global Optimization", pp.3153-3160. **(Winner of this competition)**

Tendencias actuales: Competiciones

<http://www.ntu.edu.sg/home/epnsugan/>

Benchmarks for Evaluation of Evolutionary Algorithms

We organized several competitions on benchmarking evolutionary algorithms. Recently, we also developed **several composition functions** to evaluate evolutionary algorithms. The objective of this work is explained in our Swarm Intelligence Symposium 2005 paper and also in the CEC Invited Session / Competition pages listed below.

J. J. Liang, P. N. Suganthan and K. Deb, "[Novel Composition Test Functions for Numerical Global Optimization](#)", *IEEE Swarm Intelligence Symposium*, pp. 68-75, June 2005. [Matlab codes of composition functions](#).

[CEC'05 Special Session / Competition](#) on Evolutionary Real Parameter single objective optimization

[CEC'06 Special Session / Competition](#) on Evolutionary Constrained Real Parameter single objective optimization

[CEC'07 Special Session / Competition](#) on Performance Assessment of real-parameter MOEAs

[CEC'08 Special Session / Competition](#) on large scale single objective global optimization with bound constraints

[CEC'09 Special Session / Competition](#) on Dynamic Optimization (**Primarily composition functions were used**)

[CEC09 Special Session / Competition](#) on Performance Assessment of real-parameter MOEAs

[CEC10 Special Session / Competition](#) on large-scale single objective global optimization with bound constraints

[CEC10 Special Session / Competition](#) on Evolutionary Constrained Real Parameter single objective optimization

[CEC10 Special Session on Niching](#) Introduces novel scalable test problems: B. Y. Qu and P. N. Suganthan, "Novel Multimodal Problems and Differential Evolution with Ensemble of Restricted Tournament Selection", *IEEE Congress on Evolutionary Computation*, Barcelona, Spain, July 2010.

[CEC11 Competition](#) on Testing Evolutionary Algorithms on Real-world Numerical Optimization Problems

[CEC2013 Special Session / Competition](#) on Real Parameter Single Objective Optimization

[CEC2014 Special Session / Competition](#) on Real Parameter Single Objective Optimization (**incorporates expensive function optimization**)

[CEC2014: Dynamic MOEA Benchmark Problems](#): Subhodip Biswas, Swagatam Das, P. N. Suganthan and C. A. C Coello, "[Evolutionary Multiobjective Optimization in Dynamic Environments: A Set of Novel Benchmark Functions](#)," Proc. CEC 2014, July, Beijing, China.

[CEC2015 Special Session / Competition](#) on Real Parameter Single Objective Optimization (**incorporates 3 scenarios**)

[CEC2016 Special Session / Competition](#) on Real Parameter Single Objective Optimization (**incorporates 4 scenarios**)

[CEC2017 Special Session / Competition](#) on Real Parameter Single Objective Optimization (**incorporates 3 scenarios**)

[CEC2018 Special Session / Competition](#) on Real Parameter Single Objective Optimization (**incorporates 3 scenarios**)

Milestone: CEC'2013 Real Parameter Optimization Session and Benchmark

Winners extend CMAES



•J. J. Liang, B-Y. Qu, P. N. Suganthan, Alfredo G. Hernández-Díaz, "[Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session and Competition on Real-Parameter Optimization](#)", Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, January 2013.

Special Session & Competition on Real-Parameter Single Objective Optimization at CEC-2013, Cancun, Mexico 21-23 June 2013.

Rank	Algorithm Name	Mean Ranking
1	NEIPOP ₃ CMA	0.27589
2	icmaesils	0.28289
3	DRMA-LSCh-CMA	0.30472
4	SHADE	0.32800
5	NIPOP ₃ CMA	0.34873
6	mvmo	0.36127
7	SMADE	0.45583
8	TLBSaDE	0.47042
9	DEcoblS	0.47222
10	béesrl	0.47687
11	SPSRDEMMS	0.49421
12	CMAES-RIS	0.50515
13	SPSOABC	0.51956
14	jade	0.52960
15	DE_LAPC	0.57617
16	fk-PSO	0.58058
17	TPC-GA	0.61008
18	PVADE	0.63422
19	CDASA	0.68659
20	SPSO2011	0.75352
21	PLES	0.83349

Table 1: The Table gives the mean aggregated rank of **all the 21 algorithms** ($N = 21$) across all problems and all dimensions from the CEC 2013 Special Session & Competition on Real-Parameter Single Objective Optimization after the maximum available number of function evaluations was used.

Milestone: CEC'2013 Real Parameter Optimization Session and Benchmark

Winners extend CMAES



http://www.ntu.edu.sg/home/epnsugan/index_files/cec-benchmarking.htm

http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2013/CEC2013.htm

- Benchmark Results for a Simple Hybrid Algorithm on the CEC 2013 Benchmark Set for Real parameter Optimization [#1566] . . Tianjun Liao and Thomas Stuetzle Universite Libre de Bruxelles (ULB), IRIDIA, Belgium ([Codes-Results available](#), as **ICMAES-ILS**)
- CMA-ES with Restarts for Solving CEC 2013 Benchmark Problems [#1318] . Ilya Loshchilov Ecole Polytechnique Federale de Lausanne, Laboratory of Intelligent Systems, Switzerland ([Codes-Results available](#), as **NBIPOPaCMA**)
- Dynamically updated Region Based Memetic Algorithm for the 2013 CEC Special Session and Competition on Real Parameter Single Objective Optimization [#1617] . . Benjamin Lacroix, Daniel Molina and Francisco Herrera Universidad de Granada, Spain; Universidad de Cadiz, Spain ([Codes-Results available](#), as **DRMA-LSch-CMA**)

Milestone: CEC'2014 Real Parameter Optimization Session and Benchmark

Differential Evolution winner

http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2014/CEC2014.htm



CEC2014

J. J. Liang, B-Y. Qu, P. N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, December 2013.

There are again two MAs among the three winner algorithms:

L-SHADE [1], an extension of SHADE (SHADE got the fourth place in CEC'2013)

GaAPADE [2], a hybridization of an GA, an DE and an Evolutionary Strategy;

MVMO [3]

[1] R. Tanabe, A. Fukunaga. Improving the Search Performance of SHADE Using Linear Population Size Reduction. In IEEE Congress on Evolutionary Computation, (2014), 1658-1665.

[2] S. Elsayed, S. Ruhul, D. Essam and N. Hamza. Testing United Multi-Operator Evolutionary Algorithms on the CEC2014 Real-Parameter Numerical Optimization. In IEEE Congress on Evolutionary Computation, (2014), 1650-1657.

[3] I. Erlich, J.L. Rueda, S. Wildenhues. Evaluating the Mean-Variance Mapping Optimization on the IEEE-CEC 2014 Test Suite. In IEEE Congress on Evolutionary Computation, (2014), 1625-1632.

Milestone: CEC'2014 Real Parameter Optimization Session and Benchmark

Differential Evolution winner - Código

<https://sites.google.com/site/tanaberyoji/home>

ARTÍCULO, TRANSPARENCIAS Y CÓDIGO

<https://ryojitanabe.github.io/>

<https://ryojitanabe.github.io/publication>

Success-History Based Parameter Adaptation for Differential Evolution

Ryoji Tanabe Alex Fukunaga

Graduate School of Arts and Sciences, The University of Tokyo

IEEE Congress on Evolutionary Computation
20 ~ 23, June, 2013

<https://ryojitanabe.github.io>

Shade y L-Shade

Differential Evolution (DE) [Storn and Price 97]

- Efficient method for solving numerical optimization problems
- Its search performance depends on control parameter settings

Adaptive DE

- Adaptive mechanisms for adjusting the control parameters online during the search process
- jDE [Brest 06], SaDE [Qin 09], **JADE** [Zhang 09], ...

Success-History based Adaptive DE (**SHADE**) [Tanabe 13]

- Proposed by Tanabe and Fukunaga in *Special Session 5: Differential Evolution: Past, Present and Future*
- An enhanced JADE [Zhang 09] which uses a history based parameter adaptation scheme
- The experimental results show that SHADE is competitive with previous, state of the art DE algorithms

Shade y L-Shade

The main features of JADE

- current-to- p best/1
- External archive
- Adaptive control of the F , CR parameter values

Parameter adaptation in JADE

- Each individual x_i is associated with its own CR_i and F_i
- Generates trial vectors according to these values
- At the beginning of each generation, CR_i and F_i are set probabilistically according to μ_{CR} , μ_F :
 - $CR_i = \text{NormalRand}(\mu_{CR}, 0.1)$
 - $F_i = \text{CauchyRand}(\mu_F, 0.1)$
 - μ_{CR} , μ_F : adaptive parameter and are both initialized to 0.5, and adapted during the search

Shade y L-Shade

Update rule of μ_{CR}, μ_F in JADE

- In each generation, CR_i and F_i values that succeed in generating a trial vector $\mathbf{u}_{i,G}$ which is better than the parent individual $\mathbf{x}_{i,G}$ are recorded as S_{CR}, S_F
- At the end of the generation, μ_{CR}, μ_F are updated as:

$$\mu_{CR} = (1 - c) \cdot \mu_{CR} + c \cdot \text{mean}_A(S_{CR})$$

$$\mu_F = (1 - c) \cdot \mu_F + c \cdot \text{mean}_L(S_F)$$

- c : learning rate. recommended value = 0.1
- $\text{mean}_A(S_{CR})$: Arithmetic mean of S_{CR}
- $\text{mean}_L(S_F)$: Lehmer mean of S_F
- After the update, adaptive parameter μ_{cr}, μ_F approach arithmetic or Lehmer mean of S_{CR} and S_F

Shade y L-Shade

Success-History based Adaptive DE (SHADE)

- An improved version of JADE
- Uses a different parameter adaptation mechanism based on a historical record of successful parameter settings
- Uses a historical memory M_{CR} , M_F , instead of adaptive parameter μ_{CR} , μ_F

Historical memory M_{CR} , M_F

Index	1	2	...	$H - 1$	H
M_{CR}	$M_{CR,1}$	$M_{CR,2}$...	$M_{CR,H-1}$	$M_{CR,H}$
M_F	$M_{F,1}$	$M_{F,2}$...	$M_{F,H-1}$	$M_{F,H}$

Shade y L-Shade

- The mean values of S_{CR} , S_F for each generation are stored in a historical memory M_{CR} , M_F

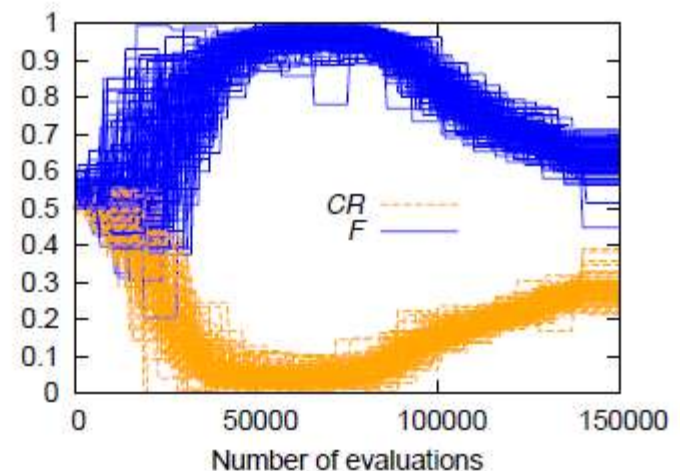
	1	2	3	...	H
M_{CR}	0.92	0.87	0.94	...	0.91
M_F	0.57	0.52	0.6	...	0.54

SHADE adapts the parameter values with diversity

- CR_i and F_i are generated by first selecting an index r_i randomly from $[1, H]$
- Example: index $r_i = 2$

$$CR_i = \text{NormalRand}(0.87, 0.1)$$

$$F_i = \text{CauchyRand}(0.52, 0.1)$$



Shade y L-Shade

Example of memory update in SHADE

1 generation				
	1	2	3	4
M_{CR}	0.5	0.5	0.5	0.5
M_F	0.5	0.5	0.5	0.5

2 generation				
	①	2	3	4
M_{CR}	0.64	0.5	0.5	0.5
M_F	0.57	0.5	0.5	0.5

4 generation				
	1	2	3	④
M_{CR}	0.64	0.64	0.73	0.23
M_F	0.57	0.6	0.62	0.13

5 generation				
	①	2	3	4
M_{CR}	0.78	0.64	0.73	0.23
M_F	0.65	0.6	0.62	0.13

- SHADE maintains a historical memory with H entries for both of the parameters CR and F , M_{CR} and M_F
- In the beginning, the contents of both memories are initialized to 0.5 and the index counter is set 1

Shade y L-Shade

Example of memory update in SHADE

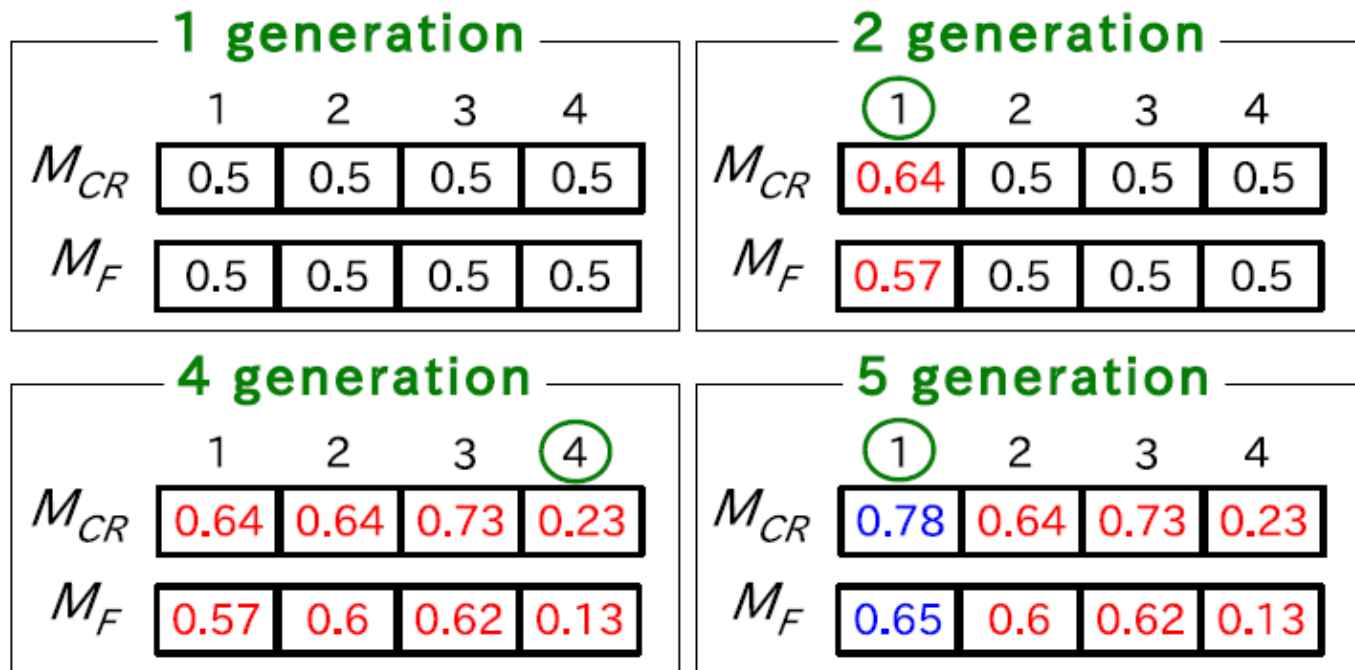
1 generation		2 generation							
	1	2	3	4		1	2	3	4
M_{CR}	0.5	0.5	0.5	0.5	M_{CR}	0.64	0.5	0.5	0.5
M_F	0.5	0.5	0.5	0.5	M_F	0.57	0.5	0.5	0.5

4 generation		5 generation							
	1	2	3	4		1	2	3	4
M_{CR}	0.64	0.64	0.73	0.23	M_{CR}	0.78	0.64	0.73	0.23
M_F	0.57	0.6	0.62	0.13	M_F	0.65	0.6	0.62	0.13

- The CR_i and F_i values used by successful individuals are recorded in S_{CR} and S_F
- At the end of the generation, the contents of memory are updated by the mean values of S_{CR} and S_F
- The index counter is incremented

Shade y L-Shade

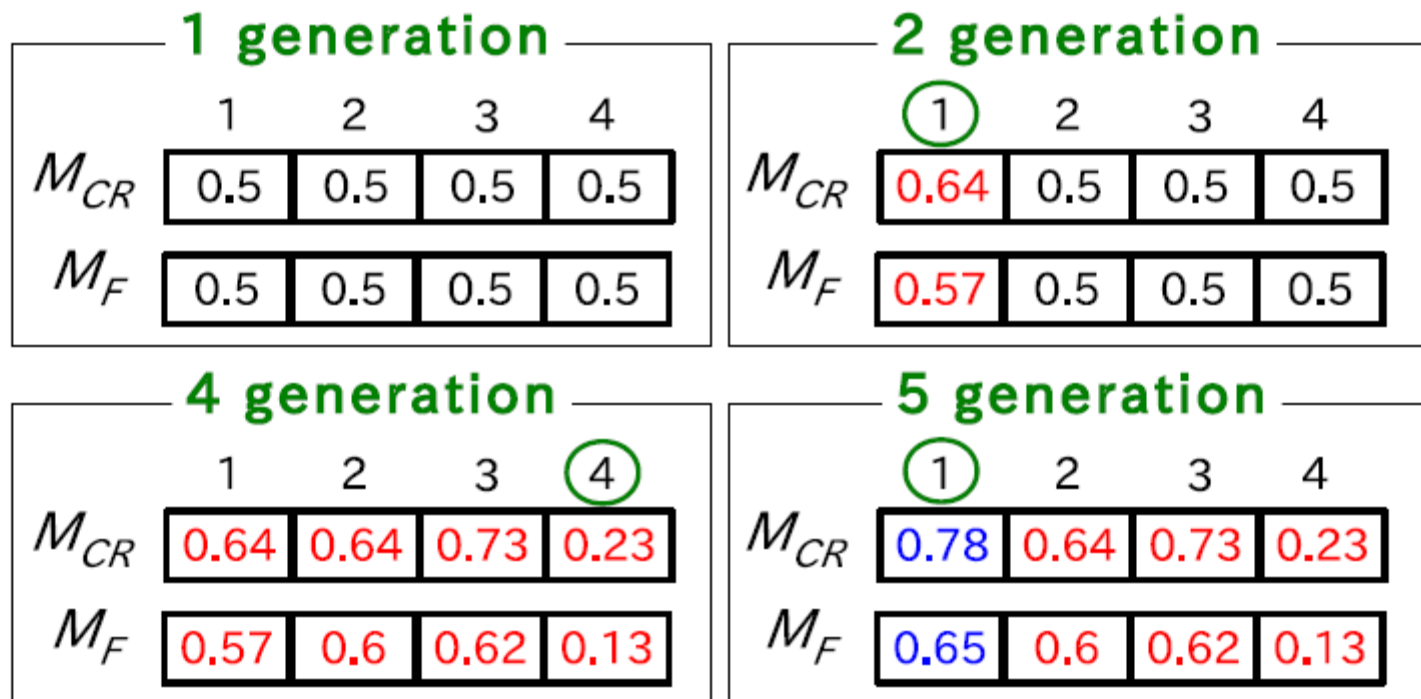
Example of memory update in SHADE



- Update procedure performed on each generation until the search finishes
- Even if S_{CR} , S_F for some particular generation contains a poor set of values, the parameters stored in memory from previous generations can not be directly, negatively impacted

Shade y L-Shade

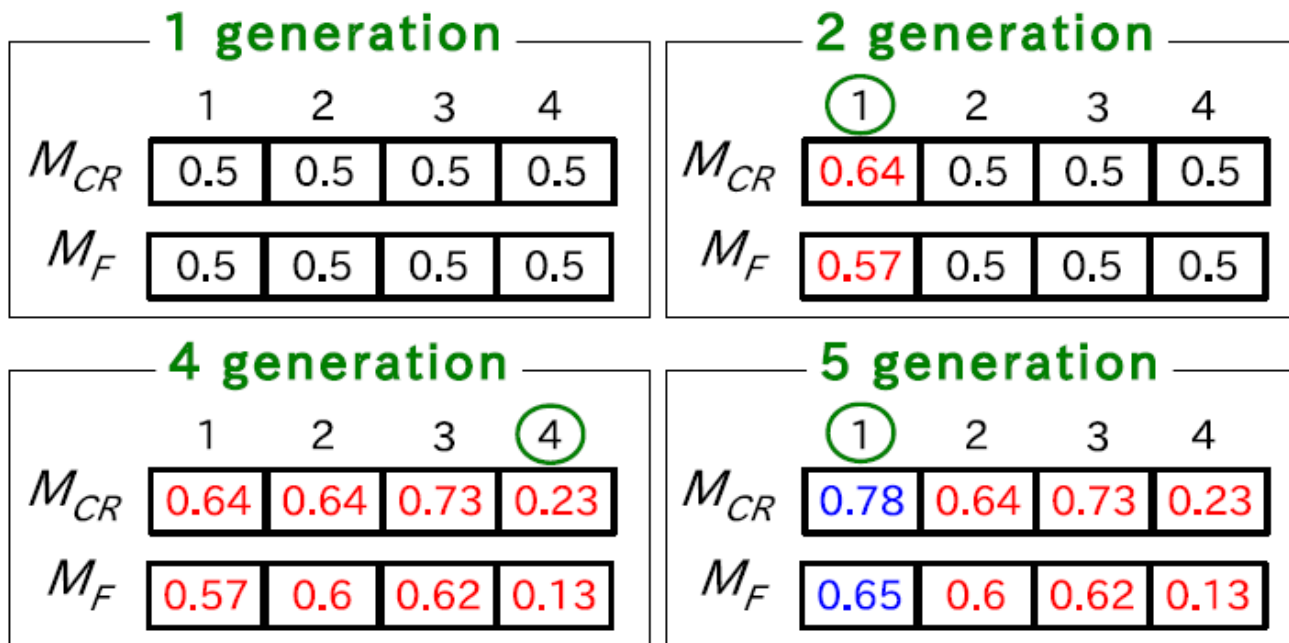
Example of memory update in SHADE



- If the index counter exceeds the memory size H ,
- The index counter wraps around to 1 again

Shade y L-Shade

Example of memory update in SHADE



The memory size H controls the rate of parameter adaptation

- Small H leads to rapid convergence of the control parameter values
- High H leads to slow convergence of the control parameter values

Shade y L-Shade

Other components of SHADE

Fitness based weighted mean for updating M_{CR} , M_F
(Section V-A)

- Taking into account of improvement of fitness $\Delta f = |f^{child} - f^{parent}|$ in memory update, the performance of SHADE can be improved

Shade y L-Shade

Fitness based weighted mean [Peng 09] for updating M_{CR} , M_F

$$M_{CR,k,G+1} = \begin{cases} \text{mean}_{WA}(S_{CR}) & \text{if } S_{CR} \neq \emptyset \\ M_{CR,k,G} & \text{otherwise} \end{cases}$$

$$M_{F,k,G+1} = \begin{cases} \text{mean}_{WL}(S_F) & \text{if } S_F \neq \emptyset \\ M_{F,k,G} & \text{otherwise} \end{cases}$$

$$\text{mean}_{WA}(S_{CR}) = \frac{\sum_{k=1}^{|S_{CR}|} w_k \cdot S_{CR,k}}{|S_{CR}|}$$

$$\text{mean}_{WL}(S_F) = \frac{\sum_{k=1}^{|S_F|} w_k \cdot F^2}{\sum_{k=1}^{|S_F|} w_k \cdot F}$$

$$w_k = \frac{\Delta f_k}{\sum_{k=1}^{|S_{CR}|} \Delta f_k}, \Delta f_k = |f_k^{child} - f_k^{parent}|$$

Why using weighted mean?

- Arithmetic mean in μ_{CR} update rule of JADE has an implicit biases to converge to a small value [Peng 09]

Shade y L-Shade

Random generation of p value in current-to- p best/1

p in current-to- p best/1

- is used to adjust the greediness of the current-to- p best/1 mutation strategy
- is static and set manually (recommended value = 0.05)

current-to- p best/1

$$\begin{aligned} \mathbf{v}_{i,G} = & \mathbf{x}_{i,G} + F_i \cdot (\mathbf{x}_{pbest,G} - \mathbf{x}_{i,G}) \\ & + F_i \cdot (\mathbf{x}_{r1,G} - \mathbf{x}_{r2,G}) \end{aligned}$$

Shade y L-Shade

Random generation of p in current-to- p best/1 (Section V-B)

- In SHADE, each individual \mathbf{x}_i has an associated p_i
- p_i is set according to the equation by generation:

$$p_i = \text{rand}[p_{min}, 0.2]$$

- $p_{min} = 2 / \text{population size}$
- The maximum value 0.2 is the maximum value of the range for p suggested by [Zhang 09]

Shade y L-Shade

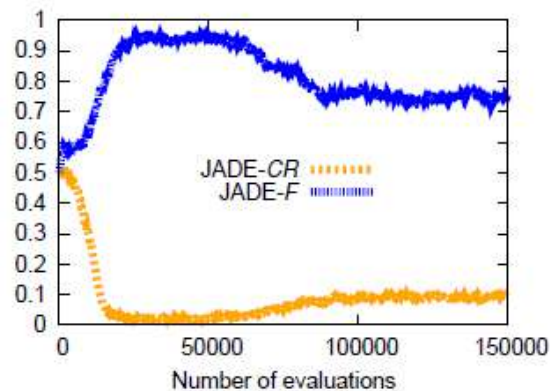
Comparison between SHADE and JADE [Zhang 09]

The control of the rate of parameter adaptation

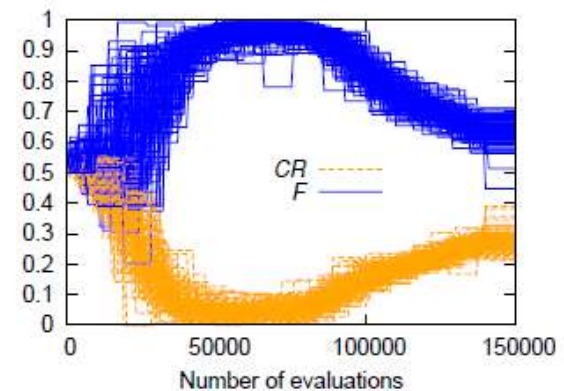
- JADE has a learning rate parameter c which controls the rate of parameter adaptation
- SHADE does not have an explicit learning rate parameter
- Instead, the memory size H plays a similar role

To guide control parameter adaptation as search progresses

JADE uses a single pair
 (μ_{CR}, μ_F)



SHADE uses a historical memory
 (M_{CR}, M_F)



Shade y L-Shade

Experimental setting

Rules of the CEC2013 benchmark competition [Liang 13]

- Dimension size $D = 10, 30, 50$
- Search space is $[-100, 100]^D$
- Error values smaller than 10^{-8} are taken as zero
- MaxFES = $D \times 10,000$
- 51 independent trials were run

SHADE Control Parameters

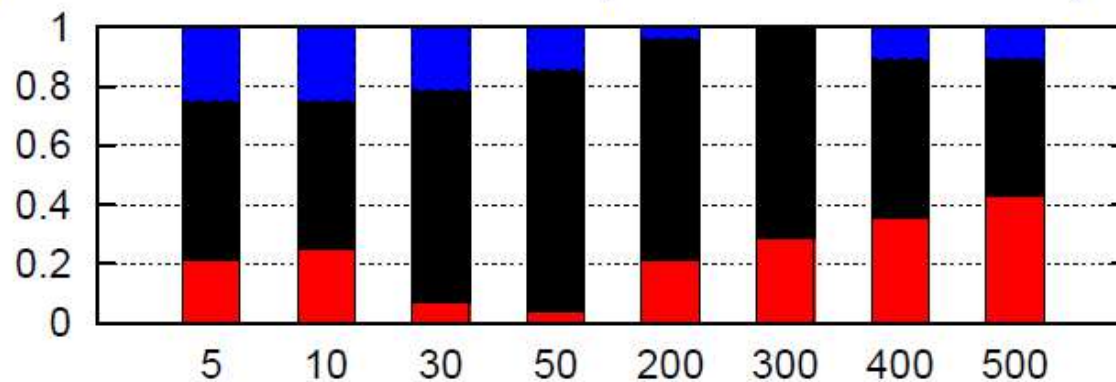
- Population size $N =$ memory size $H = 100$
- The value $N = 100$ was chosen because it was used by several other authors in recent works [Brest 06, Zhang 09]
- An evaluation of $H \in \{5, 10, 30, 50, 100, 200, 300, 400, 500\}$ confirmed that $H = 100$ resulted in good overall performance [Tanabe 13]

Shade y L-Shade

Impact of memory size H

- The memory size H needs to be set by the user
- We investigated the impact of H on the search performance of SHADE on the CEC2013 benchmarks
- $H \in 5, 10, 30, 50, 200, 300, 400, 500$
- Comparison with $H = 100$ which is the default setting

The performance of SHADE depends on the memory size H



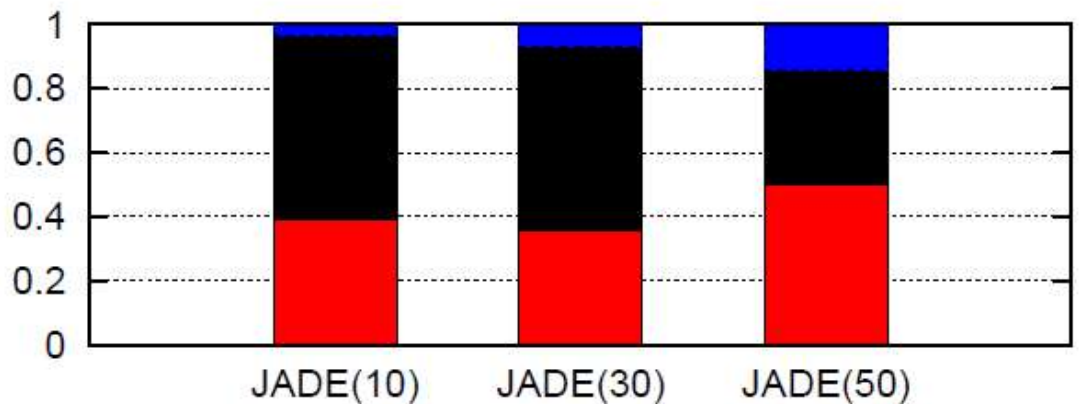
$H = 100$ was better █
No difference █
 $H = 100$ was worse █

Shade y L-Shade

Comparison with JADE on $D = 10, 30$ and 50

- Each algorithm is evaluated according to the average and standard deviation of the function error value
- Statistical significance testing: For each benchmark function, SHADE was compared to JADE using the (Wilcoxon rank-sum test, $p < 0.05$).

The percentage of better, worse, and no difference on 28 functions



SHADE was better ■ SHADE was worse ■
No difference ■

Conclusions

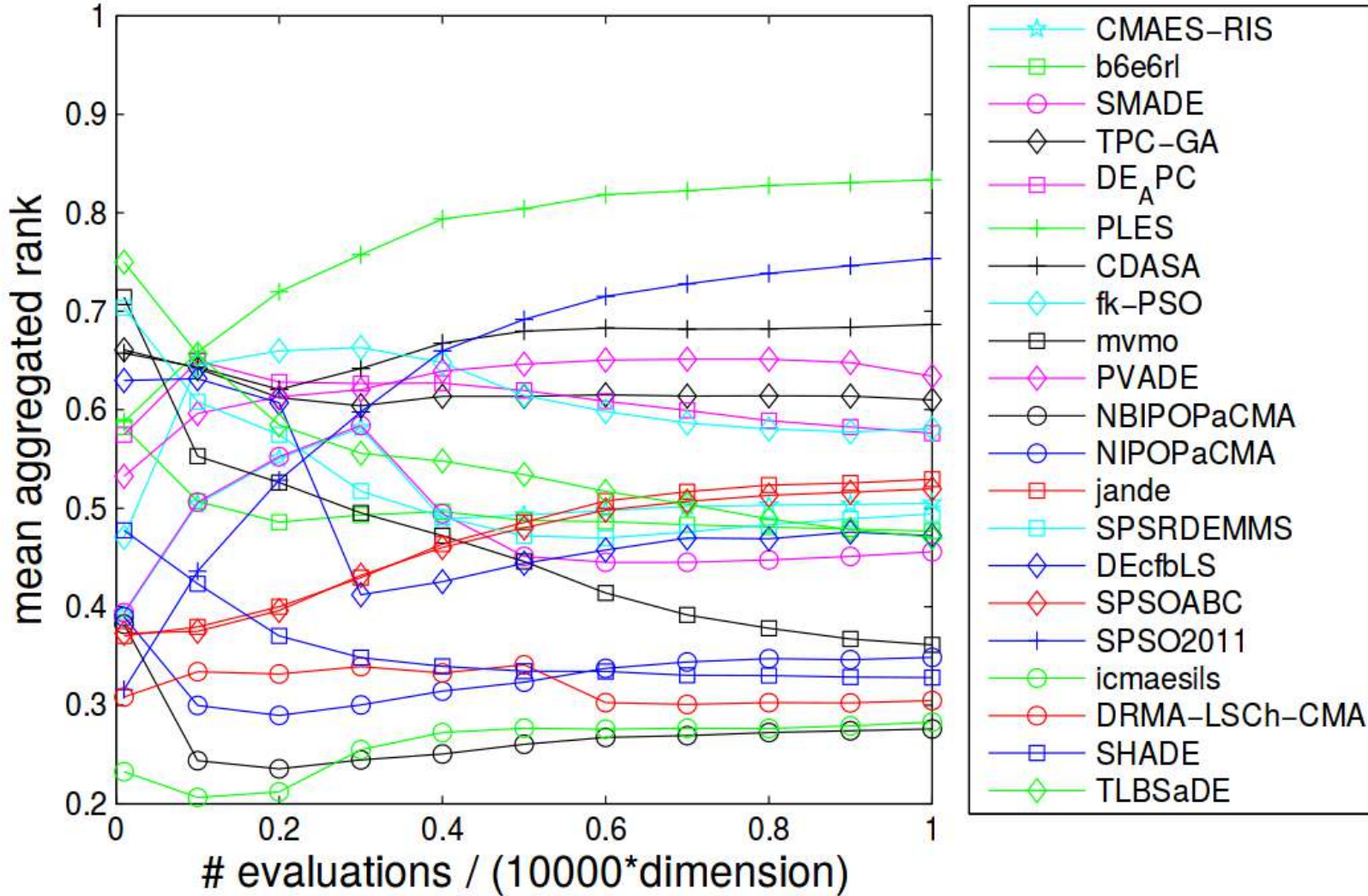
- This presentation proposed SHADE, an adaptive DE algorithm based on JADE [Zhang 09]
- SHADE uses a historical memory M_{CR} , M_F and maintains a diverse set of parameter values
- SHADE was evaluated by compared it with state-of-the-art DE algorithms on a large set of benchmark problems:
 - CEC2013 benchmarks (vs. CoDE, EPSDE, JADE, dynNP-jDE)
 - CEC2005 benchmarks (vs. CoDE)
 - 13 Classical benchmarks (vs. JADE and dynNP-jDE)
- SHADE was shown to outperform previous, state-of-the-art DE algorithms on these benchmark problems

Shade y L-Shade

Rank	Algorithm Name	Mean Ranking
1	NBIPOP ₃ CMA	0.27589
2	icmaesils	0.28289
3	DRMA-LSCh-CMA	0.30472
4	SHADE	0.32800
5	NIPOP ₃ CMA	0.34873
6	mvmo	0.36127
7	SMADE	0.45583
8	TLBS ₃ DE	0.47042
9	DEcfbLS	0.47222
10	b6e6rl	0.47687
11	SPSRDEMMMS	0.49421
12	CMAES-RIS	0.50515
13	SPSOABC	0.51956
14	jande	0.52960
15	DE_LAPC	0.57617
16	fr-PSO	0.58058
17	TPC-GA	0.61008
18	FVADE	0.63422
19	CDASA	0.68659
20	SPSO2011	0.75352
21	FLES	0.83349

Table 1: The Table gives the mean aggregated rank of **all the 21 algorithms** ($N = 21$) across all problems and all dimensions from the CEC 2013 Special Session & Competition on Real-Parameter Single Objective Optimization after the maximum available number of function evaluations was used

Shade y L-Shade



Improving the Search Performance of SHADE Using Linear Population Size Reduction

Ryoji Tanabe Alex Fukunaga

Graduate School of Arts and Sciences, The University of Tokyo, Japan

IEEE CEC-2014
12 ~ 16, July, 2014
Beijing

Shade y L-Shade

Success-History based Adaptive DE (SHADE) [Tanabe 13]

Adaptive DE

- Adapt the parameter F , CR online during the search process
- jDE [Brest 06], SaDE [Qin 09], JADE [Zhang 09], etc.

SHADE

- An improved version of JADE [Zhang 09]
- Uses a different parameter adaptation mechanism based on the success-history based adaptation

Result of SHADE in CEC 2013 competition

- 4th rank out of 21
- SHADE was outperformed by three Restart CMA-ES variants

Proposed method in this paper

- L-SHADE = SHADE + Linear population reduction method

Shade y L-Shade

Deterministic population resizing methods

- There are much work about ADE which control F, CR
- However, the population size N remains constant
- Since the optimal population size depends on the problem, introducing into adaptive N resizing in ADE is promising

Adaptive population resizing is difficult [Lobo 05]

- They tend to replace population size N with multiple, meta-level control parameters
- The meta-level control parameters need to be tuned
- Almost all adaptive resizing methods are unrealistic

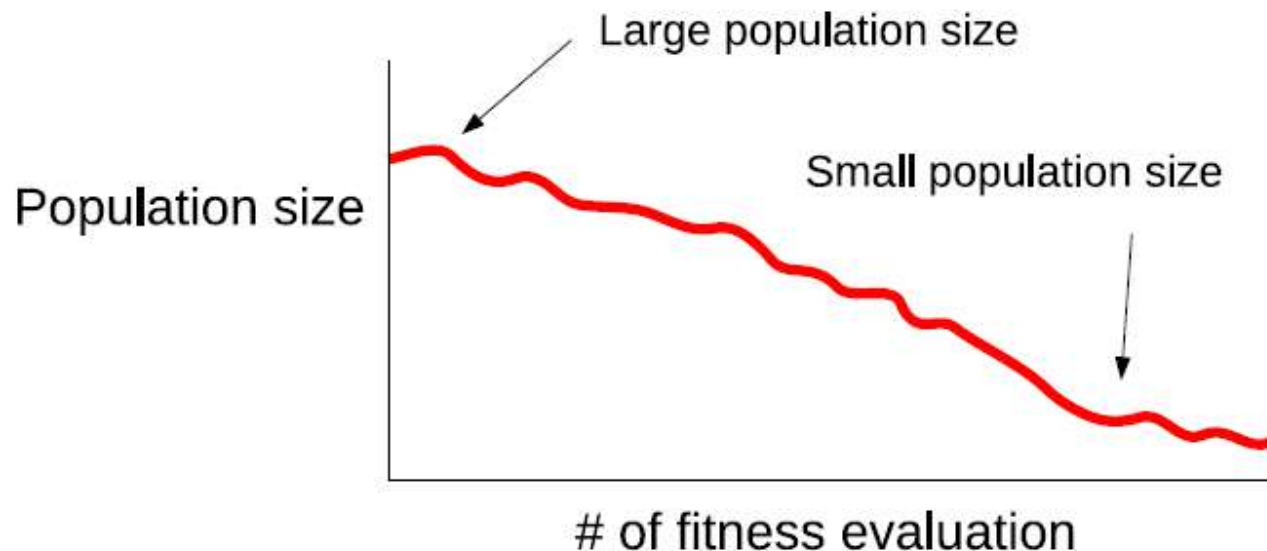
~~Adaptive~~ Deterministic population resizing methods

- Resize the population size based on simple, deterministic rules
- IPOP-CMA-ES, GL-25, IPSO, DPSR, SVPS, etc.
- They either monotonically increase or decrease the population size, based on predetermined conditions
- They are simple, but effective

Shade y L-Shade

Deterministic Population Size Reduction Strategy

- Uses a relatively large initial population
 - Good for exploring the promising regions
- Reduces its size gradually as search progresses
- As a result, the population size becomes relatively small
 - Good for exploiting the high-precision solutions



Shade y L-Shade

Three Deterministic Population Size Reduction Strategies

Dynamic Population Size Reduction (DPSR) [Brest 08]

- Reduces the population by half at predetermined intervals
- Tuning the frequency of the population reduction is hard

Simple Variable Population Sizing (SVPS) [Laredo 09]

- The shape of the population size reduction schedule is determined according to two control parameters
- Tuning the two control parameters is hard...

New?: Linear Population Size Reduction (LPSR)

- A special case of SVPS which reduces the population linearly
- Requires only initial population size
- If $N_{G+1} < N_G$ in the generation G ,
- Then, Sort individuals and delete lowest $N_G - N_{G+1}$ members

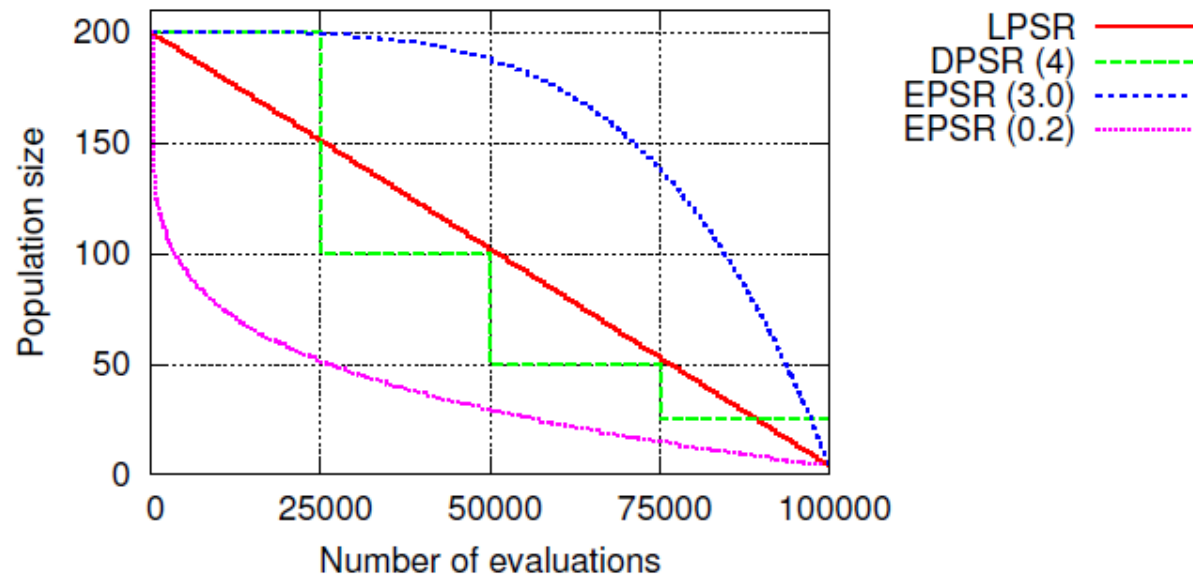
$$N_{G+1} = \text{round} \left[\left(\frac{N^{\min} - N^{\text{init}}}{MAX_NFE} \right) \cdot NFE + N^{\text{init}} \right]$$

Shade y L-Shade

Proposed method: L-SHADE = SHADE + LPSR

- At the end of generation, population size is reduced
- Population size: deterministic, F, CR : Adaptive
- Why did we use LPSR?
 - LPSR (L-SHADE) is better or similar to DPSR (D-SHADE) and Exponential PSR (E-SHADE)
 - LPSR is simpler than them

Comparison of reduction strategies



Shade y L-Shade

Parameter settings of L-SHADE in CEC'14 competition

ParamILS [Hutter 09]: iterated local search based parameter tuner

- A versatile and efficient automatic parameter tuner
- Has been shown to be highly successful in tuning algorithms

Settings of ParamILS

- Training problems: $F_{21} \sim F_{25}$ for $D = 10, 30$ in CEC'13
 - We treat CEC'14 benchmarks as unseen test problems
- Running time of ParamILS was about 4 hour

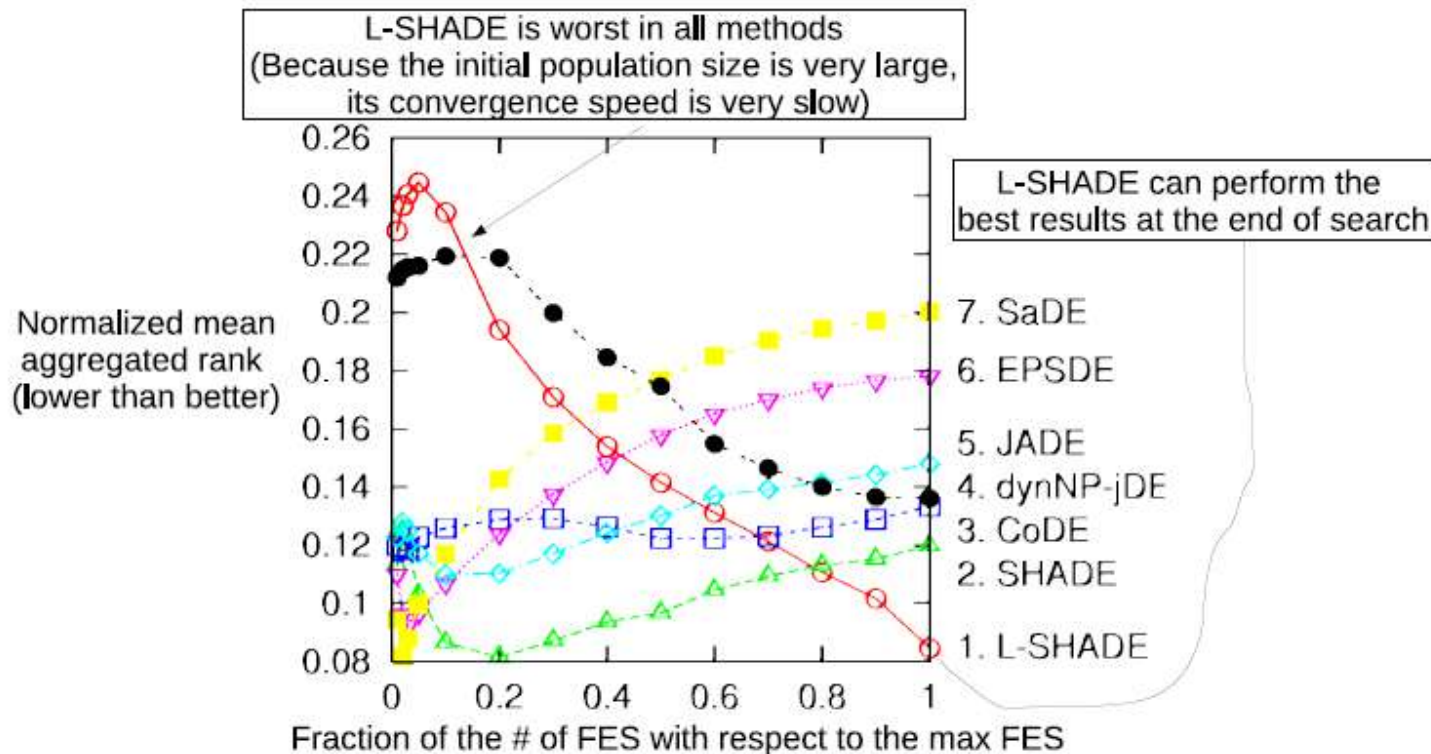
Tuned parameter settings of L-SHADE

- 1 Initial population size = $18 \times$ dimension size
 - Largest population size found in DE?
- 2 Memory size $H = 6$
- 3 Archive size = $2.6 \times$ population size
- 4 $p = 0.11$
 - Both are for current-to- p best with archive mutation

Shade y L-Shade

Compared results on CEC'14 benchmarks (vs. DE)

- The normalized mean aggregated rank [CEC 13] among the DE variants across 30 problems and 10, 30, 50 dimensions
- **L-SHADE outperforms all DE variants**
- LPSR is a promising approach for improving SHADE



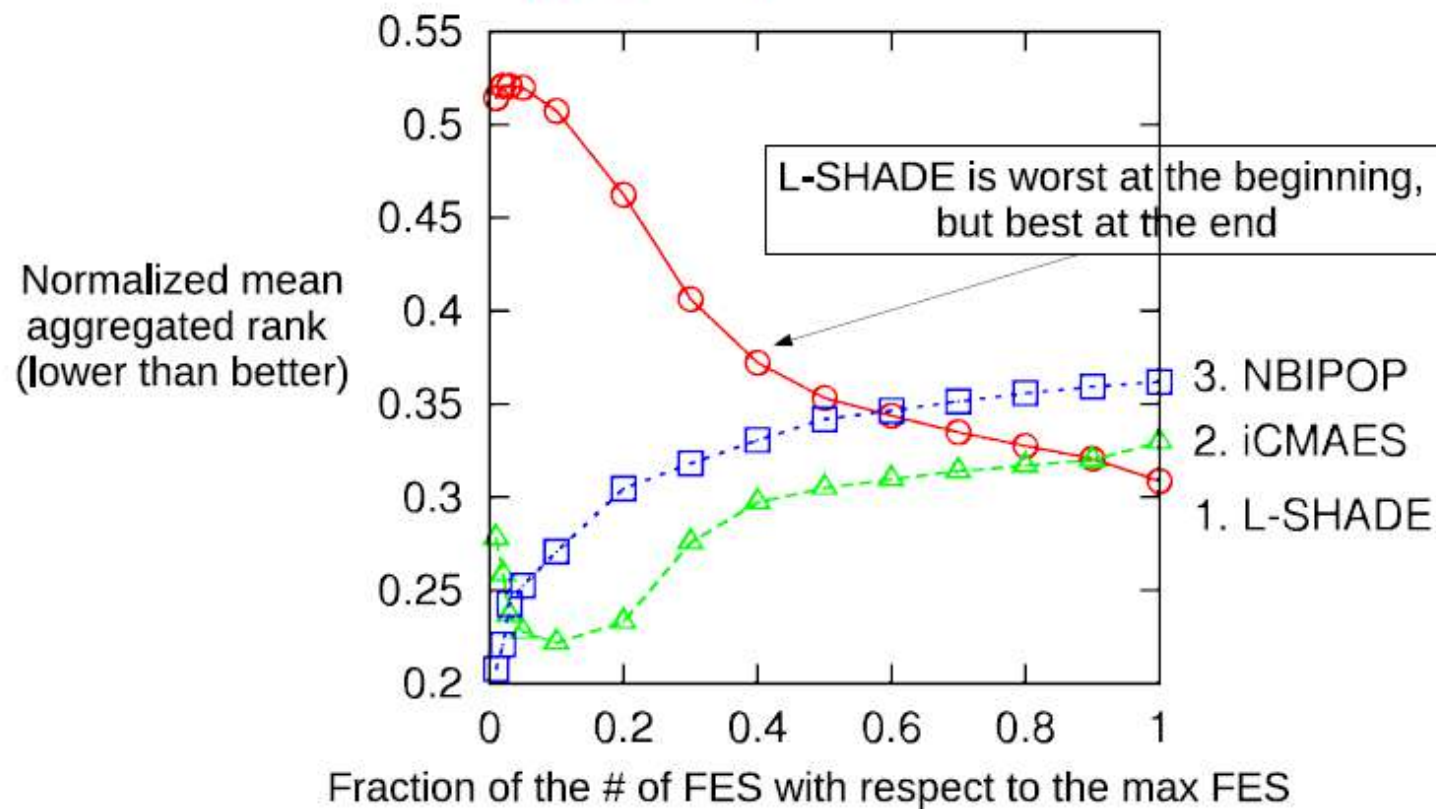
Shade y L-Shade

Compared results on CEC'14 benchmarks (vs. R-CMA-ES)

iCMAES-ILS & NBIPOP-aCMA-ES

- Both R-CMA-ES were co-winners of CEC'13 competition

L-SHADE is highly competitive with Restart CMA

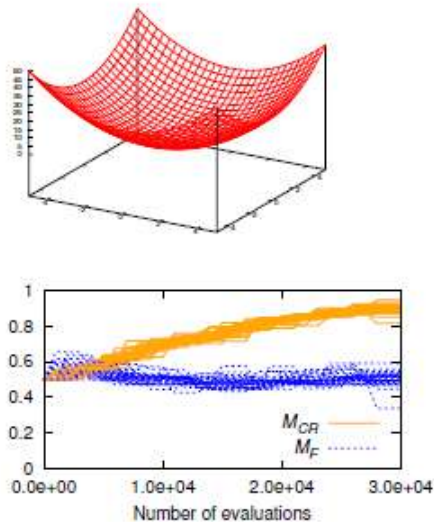


Shade y L-Shade

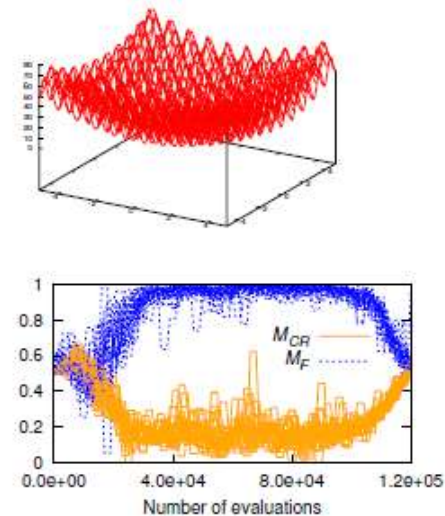
Instances of parameter adaptation process of SHADE

- F and CR values for all elements in the historical memory
 M_F and M_{CR} are shown
- SHADE maintains a diverse set of parameter values to adapt the control parameters

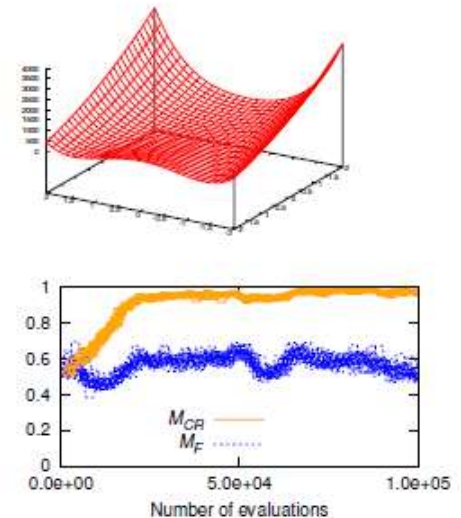
Sphere



Rastrigin



Rosenbrock



Shade y L-Shade

Summary

- We proposed L-SHADE, which is an extended SHADE introduced Linear Population Size Reduction
- The performance of L-SHADE is evaluated on CEC'14
- L-SHADE is competitive with state-of-the-art Restart CMA
- L-SHADE is simpler than Restart CMA-ES



CEC2014

L-SHADE

Improving the Search Performance of SHADE Using Linear Population Size Reduction. By Ryoji Tanabe and Alex S. Fukunaga (**Winner of the Competition**)

Winner

Milestone: CEC'2015 Real Parameter

Optimization Session and Benchmark

Differential Evolution winner

Learning Based Papers

Paper ID	Algorithm	Title
15031	SPS-L-SHADE-EIG	A Self-Optimization Approach for L-SHADE Incorporated with Eigenvector-Based Crossover and Successful-Parent-Selecting Framework on CEC 2015 Benchmark Set. Rank - #1
15096	TEBO	Tuning Maturity Model of Ecogeography-Based Optimization On CEC 2015 Single-Objective Optimization Test Problems
15170	MVMO	Testing MVMO on Learning-based Real-Parameter Single Objective Benchmark Optimization Problems Rank - #3
15230	LSHADE-ND	Neurodynamic Differential Evolution Algorithm and Solving CEC2015 Competition Problems Rank - #3
15287	ICMLSP	An Improved Covariance Matrix Learning and Searching Preference Algorithm for Solving CEC 2015 Benchmark Problems
15460	SaDPSO	A Self-adaptive Dynamic Particle Swarm Optimizer
15473	cooperation	Cooperation of Optimization Algorithms: A Simple Hierarchical Model
15485	hCC	Hybrid Cooperative Co-evolution For The CEC15 Benchmarks
15527	ABC-X-LS	A Configurable Generalized Artificial Bee Colony Algorithm with Local Search Strategies
15598	dynFWA	Dynamic Search Fireworks Algorithm for Solving CEC2015 Competition Problems
15620	DEsPA	A Differential Evolution Algorithm with Successbased Parameter Adaptation for CEC2015 Learning based Optimization Rank - #2
15642	dynFWACM	Dynamic Search Fireworks Algorithm with Covariance Mutation for Solving the CEC 2015 Learning Based Competition Problems
15667	HumanCog	HumanCog: A Cognitive Architecture for Solving Optimization Problems

Shade y L-Shade

CEC'2015 Real Parameter

Learn http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2015/CEC2015.htm

Paper ID	Algorithm	Title
15031	SPS-L-SHADE-EIG	A Self-Optimization Approach for L-SHADE Incorporated with Eigenvector-Based Crossover and Successful-Parent-Selecting Framework on CEC 2015 Benchmark Set. Rank - #1
15096	TEBO	Tuning Maturity Model of Ecogeography-Based Optimization On CEC 2015 Single-Objective Optimization Test Problems
15170	MVMO	Testing MVMO on Learning-based Real-Parameter Single Objective Benchmark Optimization Problems Rank - #3
15230	LSHADE-ND	Neurodynamic Differential Evolution Algorithm and Solving CEC2015 Competition Problems Rank - #3
15287	ICMLSP	An Improved Cov Solving CEC 2015
15460	SaDPSO	A Self-adaptive l
15473	cooperation	Cooperation of O
15485	hCC	Hybrid Cooperati
15527	ABC-X-LS	A Configurable Strategies
15598	dynFWA	Dynamic Search Fireworks Algorithm for Solving CEC2015 Competition Problems
15620	DEsPA	A Differential Evolution Algorithm with Successbased Parameter Adaptation for CEC2015 Learning based Optimization Rank - #2
15642	dynFWACM	Dynamic Search Fireworks Algorithm with Covariance Mutation for Solving the CEC 2015 Learning Based Competition Problems
15667	HumanCog	HumanCog: A Cognitive Architecture for Solving Optimization Problems

A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set

[Shu-Mei Guo](#) ; Comput. Sci. & Inf. Eng., Nat. Cheng-Kung Univ., Tainan, Taiwan ; [Jason Sheng-Hong Tsai](#) ; [Chin-Chang Yang](#) ; [Pang-Han Hsu](#)

Shade y L-Shade

CEC'2015 Real Parameter

Paper ID	Algorithm	10D	30D	50D	10D+30D+50D
15031	SPS-L-SHADE-EIG	1	1	2	1
15096	TEBO	8	9	9	9
15170	MVMO	3	3	4	3
15230	LSHADE-ND	3	3	4	3
15287	ICMLSP	12	12	12	12
15460	SaDPSO	9	8	7	8
15473	cooperation	6	7	5	7
15485	hCC	2	5	8	6
15527	ABC-X-LS	7	4	3	4
15598	dynFWA	10	11	10	10
15620	DEsPA	5	6	1	2
15642	dynFWACM	4	6	11	10



* ONLY use the mean values.
Results of 15031 and 15527 are from the papers.

SPS-L-SHADE-EIG

A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set

[Shu-Mei Guo](#) ; Comput. Sci. & Inf. Eng., Nat. Cheng-Kung Univ., Tainan, Taiwan ; [Jason Sheng-Hong Tsai](#) ; [Chin-Chang Yang](#) ; [Pang-Han Hsu](#)
[2015 IEEE Congress on Evolutionary Computation \(CEC\)](#)

A self-optimization approach and a new success-history based adaptive differential evolution with linear population size reduction (L-SHADE) which is incorporated with an eigenvector-based (EIG) crossover and a successful-parent-selecting (SPS) framework are proposed in this paper. The EIG crossover is a rotationally invariant operator which provides superior performance on numerical optimization problems with highly correlated variables. The SPS framework provides an alternative of the selection of parents to prevent the situation of stagnation. The proposed **SPS-L-SHADE-EIG** combines the L-SHADE with the EIG and SPS frameworks. To further improve the performance, the parameters of SPS-L-SHADE-EIG are self-optimized in terms of each function under IEEE Congress on Evolutionary Computation (CEC) benchmark set in 2015. The stochastic population search causes the performance of SPS-L-SHADE-EIG noisy, and therefore we deal with the noise by re-evaluating the parameters if the parameters are not updated for more than an unacceptable amount of times. The experiment evaluates the performance of the self-optimized SPS-L-SHADE-EIG in CEC 2015 real-parameter single objective optimization competition.

SPS-L-SHADE-EIG

National Cheng Kung University
Institute of Medical Informatics



成功大學醫學資訊研究所

Faculty



Shu-Mei Guo Professor

TEL: +886-6-275-7575 ext.62525

FAX: 2747076

E-mail: guosm@mail.ncku.edu.tw

Lab: **Intelligent Digital Image Processing Lab**

Research Interests

Image Processing
Medical Image Processing
Evolutionary Computing
Chaotic System
Fuzzy Theorem
System Engineering

[Enhancing Differential Evolution Utilizing Eigenvector-Based Crossover Operator](#)

[Shu-Mei Guo](#) ; [Chin-Chang Yang](#)

IEEE TEC, 19:1, 2015, 2015, Page(s):31 - 49

[Improving Differential Evolution With a Successful-Parent-Selecting Framework](#)

[Shu-Mei Guo](#) ; [Chin-Chang Yang](#) ; [Pang-Han Hsu](#) ; [Jason S. -H. Tsai](#)

IEEE TEC, 19:5, 2015, 2015,

Page(s):717 - 730

Milestone: CEC'2013-15 Real Parameter Optimization Session and Benchmark

Comparación de los mejores

Reference algorithms

IPOP-CMA-ES Winner in CEC'2005.

GAMPC Winner in the CEC'2011.

CEC'2013 Competition

ICMAES-ILS Hybrid of an ILS with CMA-ES.

NBIPOP_aCMA IPOP-CMAES with two populations.

DRMA MA with LS chaining and division in hypercubes.

CEC'2014 Competition

L-LSHADE Adaptive DE.

GaAPPAGE Hybrid GA+DE+CMA-ES.

MVMO14 MVMO.

CEC'2015 Competition

SPS-L-SHADE-EIG Replace L-SHADE with another crossover, and stuck detection.

LSHADE-ND Combine a L-SHADE with a neuro-dynamic optimization method.

Milestone: CEC'2013-15 Real Parameter Optimization Session and Benchmark

Comparación de los mejores

Results for CEC'2013

Position of algorithms by their average ranking

Competition	Algorithm	D10	D30	D50
CEC'2005	IPOP-CMAES	7	7	8
CEC'2011	GAMPC	10	10	10
CEC'2013	ICMAES-ILS	4	5	4
	NBIPOP _a CMA	9	9	9
	DRMA	5	6	6
CEC'2014	L-SHADE	1	2.5	2
	GAAPPADE	3	2.5	5
	MVMO14	8	8	7
CEC'2015	SPS-L-SHADE-EIG	2	4	3
	L-SHADE-ND	6	1	1

Milestone: CEC'2013-15 Real Parameter Optimization Session and Benchmark

Comparación de los mejores

Results for CEC'2014

Position of algorithms by their average ranking

Competition	Algorithm	D10	D30	D50
CEC'2005	IPOP-CMAES	7	5	5
CEC'2011	GAMPC	10	10	10
CEC'2013	ICMAES-ILS	4	3	1
	NBIPOP _a CMA	9	8	8
	DRMA	5	6	7
CEC'2014	L-SHADE	2	1	2
	GAAPPADE	8	7	6
	MVMO14	6	9	9
CEC'2015	SPS-L-SHADE-EIG	3	4	3
	L-SHADE-ND	1	2	4

Milestone: CEC'2013-15 Real Parameter Optimization Session and Benchmark

Comparación de los mejores

Results for CEC'2015

Position of algorithms by their average ranking

Competition	Algorithm	D10	D30	D50
CEC'2005	IPOP-CMAES	9	7	8
CEC'2011	GAMPC	10	10	10
CEC'2013	ICMAES-ILS	7	6	6
	NBIPOPaCMA	8	8	7
	DRMA	4	5	3
CEC'2014	L-SHADE	5	4	4.5
	GAAPPADE	1	2	1
	MVMO14	6	9	9
CEC'2015	SPS-L-SHADE-EIG	2	1	2
	L-SHADE-ND	3	3	4.5

Milestone: CEC'2013-15 Real Parameter Optimization Session and Benchmark

Comparación de los mejores

Conclusions

Robust algorithms

- L-SHADE and L-SHADE-ND are very robust algorithms.

Expected

- ICMAES-ILS, CEC'2013 winner, is improved by new ones.

Unexpected

- ICMAES-ILS improve CEC'2014 winners for $D=50$.
- GAAPPAGE, from CEC'2013, improve the CEC'2014 and CEC'2015 winners in CEC'2015 competition.

Results

- Older winners are unfairly ignored, improve newer ones.
- Organizers should use them as [reference algorithms](#).

Milestone: CEC'2017 Real Parameter Optimization Session and Benchmark

Papers Using CEC 2017 Bound Constrained Benchmark Set

ID	Algorithm	Paper Title
1	17315 jSO (2 nd)	Single Objective Real-Parameter Optimization Algorithm jSO
2	17321 MM_OED	Multi-method based Orthogonal Experimental Design Algorithm for Solving CEC2017 Competition Problems
3	17322 IDBestNsize	Enhanced Individual-dependent Differential Evolution with Population Size Adaptation
4	17343 RB-IPOP-CMA-ES	A Version of IPOP-CMA-ES Algorithm with Midpoint for CEC 2017 Single Objective Bound Constrained Problems
5	17051 LSHADE_SPACMA (4 th)	LSHADE with Semi-Parameter Adaptation Hybrid with CMA-ES for Solving CEC 2017 Benchmark Problems
6	17420 DES	A Differential Evolution Strategy
7	17543 DYIPO	Dynamic Yin-Yang Pair Optimization and its Performance on Single Objective Real Parameter Problems of CEC 2017
8	17544 TLBO-FL	Teaching Learning Based Optimization with Focused Learning and its Performance on CEC2017 functions
9	17447 PPSO	Proactive Particles in Swarm Optimization: a Settings-Free Algorithm for Real-Parameter Single Objective Optimization Problems
10	17260 MOS-SOCO2011/13	A comparison of three large-scale global optimizers on the CEC 2017 single objective real parameter numerical optimization benchmark
11	17106 LSHADE-cnEpSin (3 rd)	Ensemble Sinusoidal Differential Covariance Matrix Adaptation with Euclidean Neighborhood for Solving CEC2017 Benchmark Problems
12	17411 EBOwithCMAR (1 st)	Improving the local search capability of Effective Butterfly Optimizer using Covariance Matrix Adapted Retreat phase

Optimization Session and Benchmark

Cognitive Computation (2018) 10:517–544
<https://doi.org/10.1007/s12559-018-9554-0>

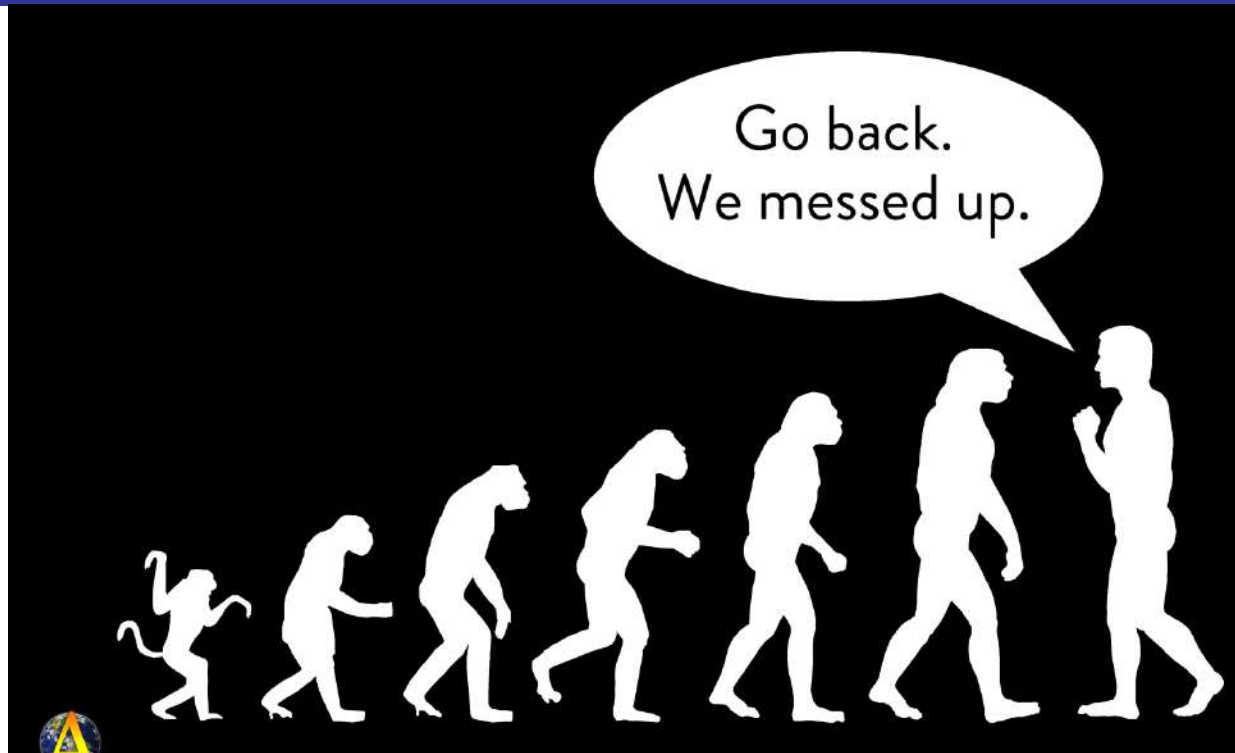


An Insight into Bio-inspired and Evolutionary Algorithms for Global Optimization: Review, Analysis, and Lessons Learnt over a Decade of Competitions

Daniel Molina¹  · Antonio LaTorre² · Francisco Herrera¹

Received: 7 August 2017 / Accepted: 3 April 2018 / Published online: 27 April 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Optimization Session and Benchmark



Summarise

- Several proposals in papers do not compare against *state-of-arts*.
- State-of-art is difficult to identify:
 - Even in competitions, there is not a clear improvement.
 - Older proposals sometimes are fairly ignored.

METAHEURÍSTICAS

TEMA 3. METAHEURÍSTICAS BASADAS EN POBLACIONES

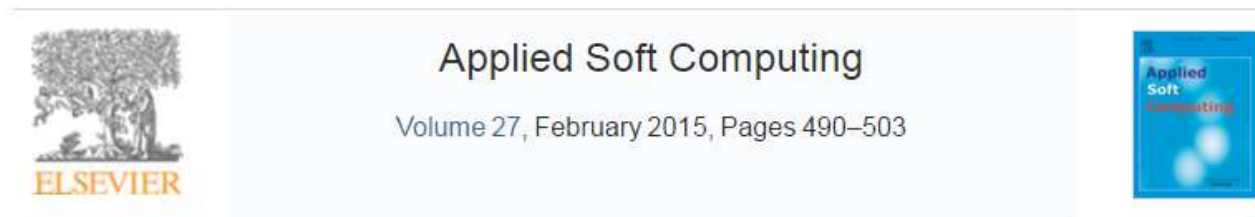
Parte II:

1. ALGORITMOS EVOLUTIVOS PARA OPTIMIZACIÓN CONTINUA I (Algoritmos genéticos)
2. EVOLUCIÓN DIFERENCIAL
3. ESTRATEGIAS DE EVOLUCIÓN
4. TEMA 6. PSO. ALGORITMOS DE NUBES DE PARTÍCULAS
5. ALGORITMOS EVOLUTIVOS PARA OPTIMIZACIÓN CONTINUA (Competiciones y modelos)
6. **NUEVOS MODELOS BIOINSPIRADOS PARA OPTIMIZACIÓN DE PARÁMETROS**

Tendencias actuales

Hay diferentes áreas de investigación que centran la atención de los investigadores en “parameter optimization”:

- Nuevos frameworks para Optimización Bioinspirada de Parámetros, y
- desarrollo de enfoques avanzados para competir con the state of the art.
- Automatic Tuning and Self-Adaptation of Algorithmic Parameters (Iterated Race and extensions)



Performance evaluation of automatically tuned continuous optimizers on different benchmark sets

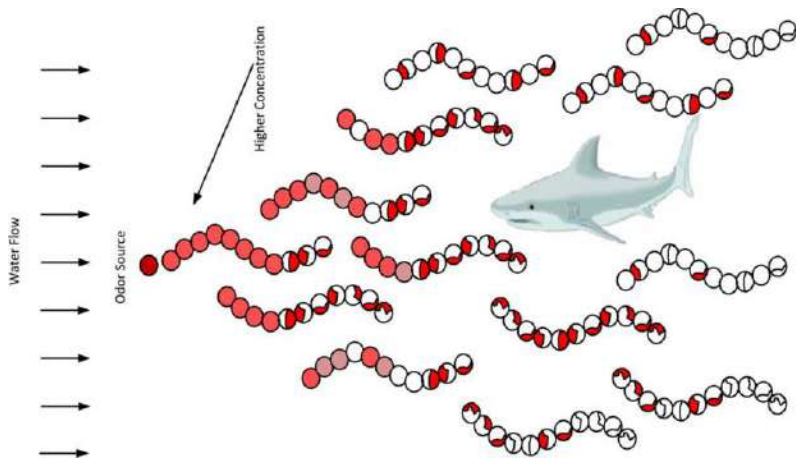
Tianjun Liao^a, Daniel Molina^b, Thomas Stützle^c

<http://www.sciencedirect.com/science/article/pii/S1568494614005584>

Tendencias actuales

Hay diferentes áreas de investigación que centran la atención de los investigadores en “parameter optimization”:

- Nuevos frameworks para Optimización Bioinspirada de Parámetros, y
- desarrollo de enfoques avanzados para competir con the state of the art.



Shark smell optimization



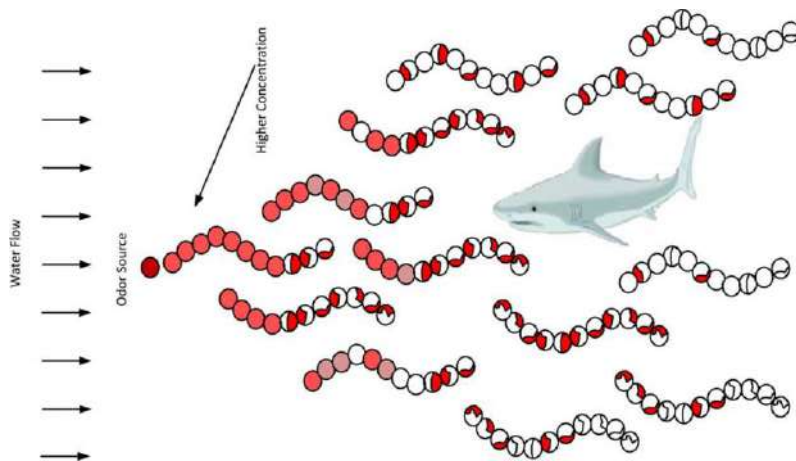
Grey Wolf Optimization

Nuevos modelos bioinspirados para la optimización de parámetros (SSO, 2016)

Shark smell optimization

O. Avedinia, N. Amdjady, A. Chasemi (2014). **A new metaheuristic algorithm based Shart Smell Optimization.** Complexity, 21: 97–116, 2016

Movement of Shark Toward Prey



A New Metaheuristic Algorithm Based on Shark Smell Optimization

OVEIS ABEDINIA,¹ NIMA AMJADY,¹ AND ALI GHASEMI²

¹Department of Electrical Engineering, Semnan University, Semnan, Iran and ²Young Researchers and Elite Club, Ardebil Branch, Islamic Azad University, Ardebil, Iran

$$|v_{ij}^k| = \min \left[\left| \eta_k \cdot R1 \cdot \frac{\partial(\text{OF})}{\partial x_j} \Big|_{x_{ij}^k} + \alpha_k \cdot R2 \cdot v_{ij}^{k-1} \right|, \left| \beta_k \cdot v_{ij}^{k-1} \right| \right]$$

$$j = 1, \dots, \text{ND}, \quad i = 1, \dots, \text{NP}, \quad k = 1, \dots, k_{\max}$$

$$Y_i^{k+1} = X_i^k + V_i^k \cdot \Delta t_k \quad i = 1, \dots, \text{NP} \quad k = 1, \dots, k_{\max} \quad (9)$$

where Δt_k indicates time interval of the stage k . For simplicity, it is assumed that $\Delta t_k = 1$ for all stages.

From optimization viewpoint, shark implements a local search in each stage to find better candidate solutions. This local search is modeled in the SSO algorithm as follows:

$$Z_i^{k+1,m} = Y_i^{k+1} + R3 \cdot Y_i^{k+1} \quad (10)$$

$$m = 1, \dots, M \quad i = 1, \dots, \text{NP} \quad k = 1, \dots, k_{\max}$$

$$X_i^{k+1} = \arg \max \{ \text{OF}(Y_i^{k+1}), \text{OF}(Z_i^{k+1,1}), \dots, \text{OF}(Z_i^{k+1,M}) \}$$

$$i = 1, \dots, \text{NP}$$

Nuevos modelos bioinspirados para la optimización de parámetros - Modelos

Advances in Engineering Software 69 (2014) 46–61



Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advensoft



Grey Wolf Optimizer

Seyedali Mirjalili^{a,*}, Seyed Mohammad Mirjalili^b, Andrew Lewis^a

^a School of Information and Communication Technology, Griffith University, Nathan Campus, Brisbane QLD 4111, Australia

^b Department of Electrical Engineering, Faculty of Electrical and Computer Engineering, Shahid Beheshti University, G.C. 1983963113, Tehran, Iran



Information Sciences 329 (2016) 597–618



Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins



**Estudio
comparativo
incluyendo
CMAES**

Across neighborhood search for numerical optimization

Guohua Wu^{1,*}

Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, Hunan, PR China



Nuevos modelos bioinspirados para la optimización

de parámetros (Jaya: Sanskrit word meaning victory, 2016)

If $X_{j,k,i}$ is the value of the j^{th} variable for the k^{th} candidate during the i^{th} iteration, then this value is modified as per the following Eq. (1).

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i} (X_{j,best,i} - |X_{j,k,i}|) - r_{2,j,i} (X_{j,worst,i} - |X_{j,k,i}|) , \quad (1)$$

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i} (X_{j,best,i} - |X_{j,k,i}|) - r_{2,j,i} (X_{j,worst,i} - |X_{j,k,i}|) ,$$

where, $X_{j,best,i}$ is the value of the variable j for the *best* candidate and $X_{j,worst,i}$ is the value of the variable j for the *worst* candidate. $X'_{j,k,i}$ is the updated value of $X_{j,k,i}$ and $r_{1,j,i}$ and $r_{2,j,i}$ are the two random numbers for the j^{th} variable during the i^{th} iteration in the range $[0, 1]$. The term “ $r_{1,j,i} (X_{j,best,i} - |X_{j,k,i}|)$ ” indicates the tendency of the solution to move closer to the best solution and the term “ $-r_{2,j,i} (X_{j,worst,i} - |X_{j,k,i}|)$ ” indicates the tendency of the solution to avoid the worst solution. $X'_{j,k,i}$ is accepted if it gives better function value. All the accepted function values at the end of iteration are maintained and these values become the input to the next iteration.

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems

Nuevos modelos bioinspirados para la optimización de parámetros (Cuckoo Search, 2009)

Cuckoo Search via Lévy Flights

Xin-She Yang
Department of Engineering,
University of Cambridge
Trumpinton Street
Cambridge CB2 1PZ, UK

Suash Deb
Department of Computer Science & Engineerin
C. V. Raman College of Engineering
Bidyanagar, Mahura, Janla
Bhubaneswar 752054, INDIA

Citation detail: X.-S. Yang, S. Deb, "Cuckoo search via Lévy flights", in: *Proc. World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, December 2009, India. IEEE Publications, USA, pp. 210-214 (2009).

Applied Soft Computing 11 | 2011 | 5508-5518

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc



Cuckoo Optimization Algorithm

Ramin Rajabioun*

Control and Intelligent Processing Centre of Excellence (CIPCE), School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

Cuckoo Search via Lévy Flights

begin

Objective function $f(\mathbf{x})$, $\mathbf{x} = (x_1, \dots, x_d)^T$

Generate initial population of

n host nests \mathbf{x}_i ($i = 1, 2, \dots, n$)

while ($t < \text{MaxGeneration}$) or (*stop criterion*)

Get a cuckoo randomly by Lévy flights

evaluate its quality/fitness F_i

Choose a nest among n (say, j) randomly

if ($F_i > F_j$),

replace j by the new solution;

end

A fraction (p_a) of worse nests

are abandoned and new ones are built;

Keep the best solutions

(or nests with quality solutions);

Rank the solutions and find the current best

end while

Postprocess results and visualization

end

Figure 1: Pseudo code of the Cuckoo Search (CS).

cuckoo i , a Lévy flight is performed

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \alpha \oplus \text{Lévy}(\lambda), \quad (1)$$



Metaheuristics—the metaphor exposed

Kenneth Sörensen

University of Antwerp, Operations Research Group ANT/OR, Prinsstraat 13 – B5xx, 2000 Antwerp, Belgium

E-mail: kenneth.sorensen@ua.ac.be [Sörensen]

Received 2 November 2012; received in revised form 12 November 2012; accepted 13 November 2012

Abstract

In recent years, the field of combinatorial optimization has witnessed a true tsunami of “novel” metaheuristic methods, most of them based on a metaphor of some natural or man-made process. The behavior of virtually any species of insects, the flow of water, musicians playing together – it seems that no idea is too far-fetched to serve as inspiration to launch yet another metaheuristic. In this paper, we will argue that this line of research is threatening to lead the area of metaheuristics away from scientific rigor. We will examine the historical context that gave rise to the increasing use of metaphors as inspiration and justification for the development of new methods, discuss the reasons for the vulnerability of the metaheuristics field to this line of research, and point out its fallacies. At the same time, truly innovative research of high quality is being performed as well. We conclude the paper by discussing some of the properties of this research and by pointing out some of the most promising research avenues for the field of metaheuristics.

Keywords: optimization; combinatorial optimization; metaheuristics; heuristics

Nuevos modelos bioinspirados para la optimización de parámetros –

Estado del arte

Revisión de propuestas de Metaheurísticas

<https://arxiv.org/abs/2002.08136>

Published: 05 July 2020

Comprehensive Taxonomies of Nature- and Bio-inspired Optimization: Inspiration Versus Algorithmic Behavior, Critical Analysis Recommendations

[Daniel Molina](#) , [Javier Poyatos](#), [Javier Del Ser](#), [Salvador García](#), [Amir Hussain](#) & [Francisco Herrera](#)

Cognitive Computation **12**, 897–939(2020) | [Cite this article](#)

6 Citations | 0 Altmetric | [Metrics](#)

arXiv.org > cs > arXiv:2002.08136

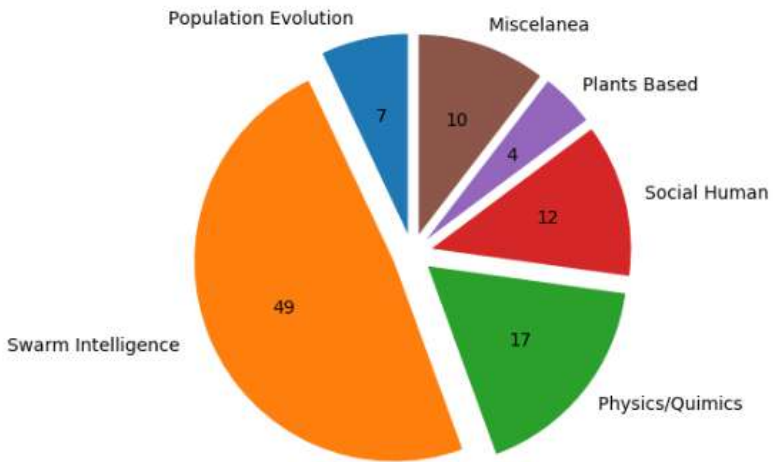
Computer Science > Artificial Intelligence

Comprehensive Taxonomies of Nature- and Bio-inspired Optimization: Inspiration versus Algorithmic Behavior, Critical Analysis and Recommendations

[Daniel Molina](#), [Javier Poyatos](#), [Javier Del Ser](#), [Salvador García](#), [Amir Hussain](#), [Francisco Herrera](#)

(Submitted on 19 Feb 2020 (v1), last revised 20 Feb 2020 (this version, v2))

In recent years, a great variety of nature- and bio-inspired algorithms has been reported in observed in Nature in order to efficiently address complex optimization problems. In the last few years, this field has grown considerably, reaching unprecedented levels that dark the future prospects of this field. This paper presents a comprehensive, principle-based taxonomies that allow researchers to organize existing algorithms according to different criteria: the source of inspiration and the behavior of each algorithm. Using these taxonomies, we analyze similarities between them, and the identification of the most similar classical algorithm for each category. Furthermore, what is claimed in their public disclosure: specifically, we show that more than one-third of the algorithms are grounded on the conclusions of our critical analysis, we give several recommendations as a growing research field.



Nuevos modelos bioinspirados para la optimización de parámetros –

Estado del arte

Revisión de propuestas de Me

<https://arxiv.org/abs/2002.08136>

arXiv.org > cs > arXiv:2002.08136

Computer Science > Artificial Intelligence

Comprehensive Taxonomies of Algorithmic Behavior, Critical

Daniel Molina, Javier Poyatos, Javier Del Ser, S

(Submitted on 19 Feb 2020 (v1), last revised 20 Feb 2020

In recent years, a great variety of nature- and bio-inspired algorithms have been observed in Nature in order to efficiently address con- siderable growth, reaching unprecedented levels of complexity. This paper presents comprehensive, principle-based taxonomies that allow for the classification of different criteria: the source of inspiration and the behavior of the algorithms, and the identification of the similarities between them.

Optimización inspirada en la naturaleza y en la biología: lo bueno, lo malo, lo feo y lo esperanzador

D. Molina, J Poyatos, E Osaba, J DEL SER, F Herrera DYNA 97 (2), 114-117, 2022

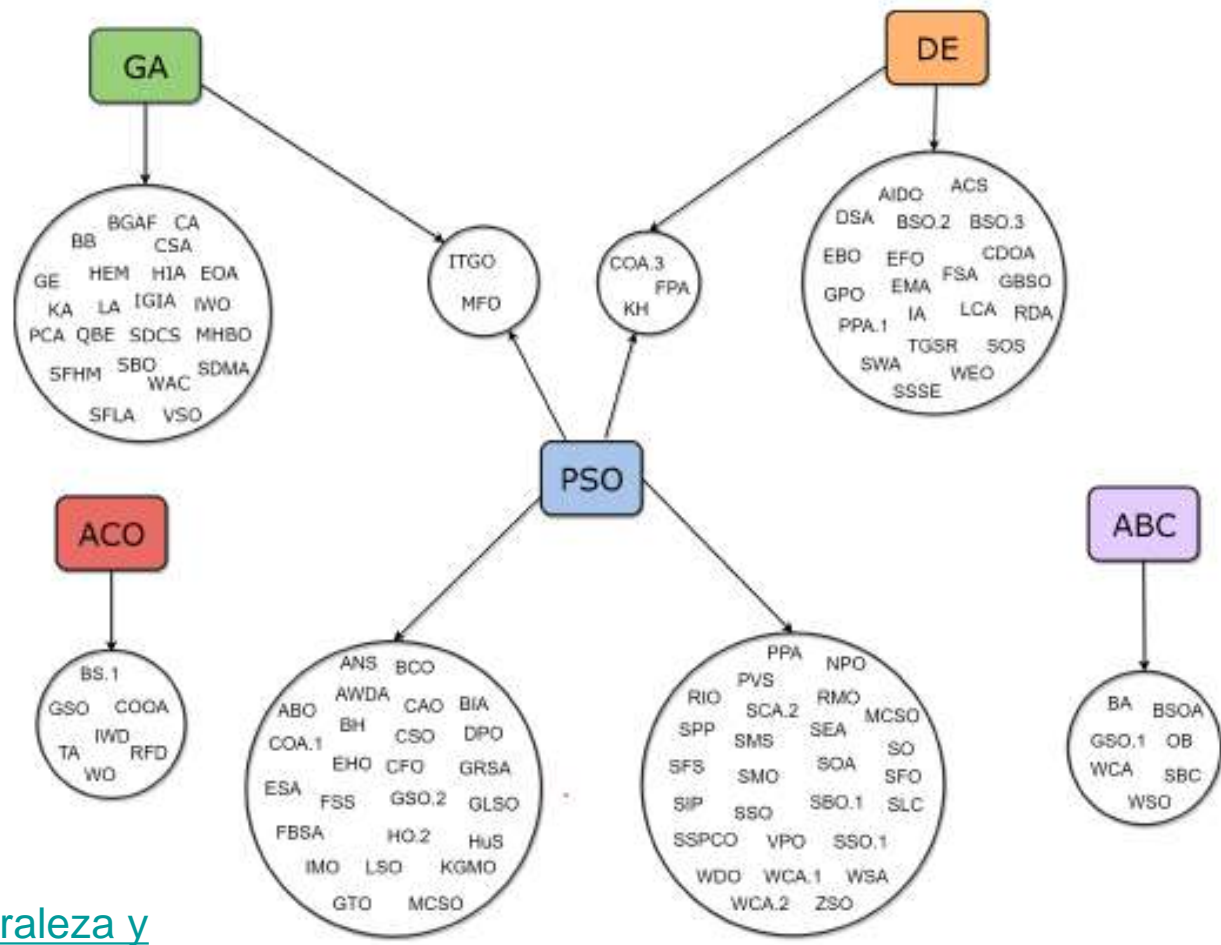


Figure 6: Classification of proposals by its original algorithm.

PRACTICA 4 – Hasta 5 ptos – Alternativa

Examen (1.5 + 2.0 + 1.5)

- Elegir una MH (MHe) no estudiada en clase (Hasta 1.5)**
(Alternativamente hacer una propuesta original)
 - Resumen (hasta 2-3 paginas)**
 - Presentación transparencias.** Transparencia: analizar componentes que provocan exploración vs componentes que provocan explotación. Analizar este equilibrio.
- Adaptar el uso de la MH a uno de los problemas de prácticas. Discutir cómo y hacer un estudio experimental comparando con los resultados de las prácticas 1 a 3 que se tengan. (Hasta 1) (Hasta 1.5 por dificultad)**
- Diseñar una propuesta de calidad (mejorando a MHe) Algoritmo híbrido memético (MHe + Local Search, opción u otra hibridación) (Hasta 1)**
- Plantear mejoras sobre el comportamiento/diseño de MHe (Hasta 1.5 con estudio experimental):**
 - Modificar/introducir componentes para mejorar el equilibrio diversidad-convergencia (ej. CHC, GADEGD, ...)**
 - Adaptar el algoritmo para que obtenga múltiples soluciones.**

PRACTICA 4 – Hasta 5 ptos

Alternativa Examen

- Usaremos benchmark “CEC 2017 Special Session and Competition on Single Objective Bound Constrained Real-Parameter Numerical Optimization”.
- 30 funciones, distinto nivel de complejidad.
- Distintos valores de dimensión: 10, 30, 50 (no 100).
- Criterio de parada: $10.000 * dimension$.
- Rango de todas las funciones: $[-100, 100]$.
- Se ejecuta cada una X veces para calcular la media (usamos $T=10$, por reducir tiempos, el estándar son 50).
- Se mide el error con respecto al óptimo ($fun * 100$): 100 para f_1 , 200 f_2 , ...
- Se calcula el error cada cierto % de evaluaciones: 1%, 2%, ..., 10%, 20%, ...

G. Wu, R. Mallipeddi, P. N. Suganthan, Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization, Tech. rep., Nanyang Technological University, Singapore (2016).

3/16

PRACTICA 4 – Hasta 5 ptos

Alternativa Examen

- Código actualizado en <https://github.com/dmolina/cec2017real/>.
- Disponible en C++, con wrapper para Python.
- Se puede compilar como librería, o dentro del proyecto.
- Ficheros:
 - *cec17_test_func.c*: fichero original del benchmark.
 - *cec17.c*: con las funciones del API simplificado.
 - *input_data*: debe estar en el directorio donde se ejecute el binario.
 - *extract.py*: script en Python para agrupar los resultados.

PRACTICA 4 – Hasta 5 ptos

Alternativa Examen

A) Proponer una MH nueva inspirada en algún fenómenos de la naturaliza (pasos 1 a 3)

B) Analizar una MH reciente (Mhe) <https://arxiv.org/abs/2002.08136>

1. Parte I. (Hasta 1.0)

A) Resumen (hasta 2-3 paginas)

B) Presentación transparencias

2. Parte II. (Hasta 1.5 ptos) Trabajar con la competición CEC2017 y utilizar el software presentado por Daniel Molina para el análisis de resultados <https://taicolab.org/>.

Select Benchmark

Select the benchmark used for comparison:

CEC'2017 Real-Parameter Optimization

Benchmark for the Real-Parameter Optimization competitions.

Select the dimension value:

PRACTICA 4 – Hasta 5 ptos

Alternativa Examen

- 3. Diseñar una propuesta de mejora con una hibridación Ej. Algoritmo híbrido memético (MHe + Local Search, u otra hibridación) (Hasta 1 pto.), y hacer el estudio experimental comparándolo con el software <https://tacolab.org/>.**
- 4. Plantear mejoras sobre el comportamiento/diseño de la MHe (Hasta 1.5 ptos. con estudio experimental)**

Hay que entregar una memoria completa de la práctica, tal como se hace en las prácticas de clase.

PRACTICA 4 – Hasta 5 ptos

Alternativa Examen

Cuestiones

- ¿Cuáles son los criterios que se van a utilizar a la hora de calificar la práctica alternativa?

Se tienen 4 bloques y en cada uno se evaluará el trabajo realizado, que se ajuste a lo solicitado y la calidad del estudio experimental y su análisis. En el bloque 1 se analiza la calidad de la descripción, en los bloques 2 y 3 el estudio experimental y análisis, y en el bloque 4 se tienen en cuenta originalidad y la calidad de la propuesta. Si la MHe es original, se tendrá en cuenta la creatividad asociada.

- ¿Hay que implementar nuestra metaheurística inventada, o es suficiente con una explicación mediante pseudocódigo?

Hay que implementarla y analizarla experimentalmente con la competición CEC2017, así como el modelo memético (punto 3) y las extensiones (punto 4) de la propuesta.

PRACTICA 4 – Hasta 5 ptos

Alternativa Examen

Cuestiones

- ¿Hay que superar alguna metaheurística tomada como referencia?

No necesariamente, la naturaleza de la MHe condicionará la calidad del resultado.

- ¿Podrían subir algún guión especificando qué hay que entregar exactamente?

Este documento sirve de guión, junto en el fichero pdf de Daniel Molina que explica cómo hacer la experimentación.
Es necesario entregar un documento que describa con detalle el estudio realizado y entregar el software asociado.

PRACTICA 4 – Hasta 5 ptos

Alternativa Examen

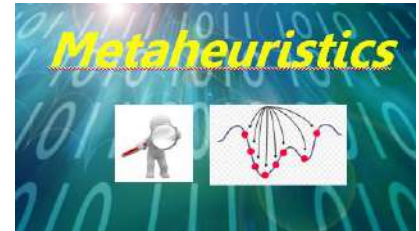
Cuestiones

- ¿Cuándo hay que avisarle (si es que hace falta) de cuál será nuestra metaheurística inventada?

Para indicar que MHe se va a utilizar (original o del documento mencionado) se tiene plazo hasta final de Mayo, y se necesita confirmación del profesor para evitar un uso excesivo de la misma MHe. Puede no entregarse la práctica e ir al examen, avisando con 48 horas de antelación a la fecha del examen.

METAHEURÍSTICAS

2021-2022



- Tema 1. Introducción a las Metaheurísticas
- Tema 2. Modelos de Búsqueda: Entornos y Trayectorias vs Poblaciones
- Tema 3. Metaheurísticas Basadas en Poblaciones
- Tema 4: Algoritmos Meméticos
- Tema 5. Metaheurísticas Basadas en Trayectorias
- Tema 6. Metaheurísticas Basadas en Adaptación Social
- Tema 7. Aspectos Avanzados en Metaheurísticas
- Tema 8. Metaheurísticas Paralelas