

BIOINFORMÁTICA

2013 - 2014

PARTE I. INTRODUCCIÓN

- Tema 1. Computación Basada en Modelos Naturales

PARTE II. MODELOS BASADOS EN ADAPTACIÓN SOCIAL (Swarm Intelligence)

- Tema 2. Introducción a los Modelos Basados en Adaptación Social
- Tema 3. Optimización Basada en Colonias de Hormigas
- Tema 4. Optimización Basada en Nubes de Partículas (Particle Swarm)

PARTE III. COMPUTACIÓN EVOLUTIVA

- Tema 5. Introducción a la Computación Evolutiva
- Tema 6. Algoritmos Genéticos I. Conceptos Básicos
- Tema 7. Algoritmos Genéticos II. Diversidad y Convergencia
- Tema 8. Algoritmos Genéticos III. Múltiples Soluciones en Problemas Multimodales
- Tema 9. Estrategias de Evolución y Programación Evolutiva
- **Tema 10. Algoritmos Basados en Evolución Diferencial (Differential Evolution – DE)**
- Tema 11. Modelos de Evolución Basados en Estimación de Distribuciones (EDA)
- Tema 12. Algoritmos Evolutivos para Problemas Multiobjetivo
- Tema 13. Programación Genética
- Tema 14. Modelos Evolutivos de Aprendizaje

PARTE IV. OTROS MODELOS DE COMPUTACIÓN BIOINSPIRADOS

- Tema 15. Sistemas Inmunológicos Artificiales
- Tema 16. Otros Modelos de Computación Natural/Bioinspirados

COMPUTACIÓN EVOLUTIVA

PARADIGMAS DE EVOLUCIÓN

COMPUTACIÓN EVOLUTIVA

PARADIGMAS CLÁSICOS DE EVOLUCIÓN

ALGORITMOS GENÉTICOS

PROGRAMACIÓN GENÉTICA

ESTRATEGIAS DE EVOLUCIÓN

PROGRAMACIÓN EVOLUTIVA

OTROS MODELOS DE EVOLUCIÓN DE POBLACIONES

BASADOS EN MODELOS PROBABILÍSTICOS: BSC, PBIL, EDA, ...

PARTICLE SWARM: ADAPTACIÓN SOCIAL

BÚSQUEDA DISPERSA
(scatter search)

DIFFERENTIAL EVOLUTION

ALGORITMOS MEMÉTICOS

BIOINFORMÁTICA

TEMA 10: ALGORITMOS BASADOS EN EVOLUCIÓN DIFERENCIAL (Differential Evolution – DE)

1. EVOLUCIÓN DIFERENCIAL
2. VARIANTES DE LA EVOLUCIÓN DIFERENCIAL
3. ALGUNOS TRABAJOS 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, 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

Initialization

A population $P_{x,0}$ of Np D -dimensional parameter vectors $\mathbf{x}_{i,0}=[x_{1,i,0},\dots,x_{D,i,0}]$, $i=1,\dots,Np$ is randomly generated within the prescribed lower and upper bound $\mathbf{b}_L=[b_{1,L},\dots,b_{D,L}]$ and $\mathbf{b}_U=[b_{1,U},\dots,b_{D,U}]$

Example: the initial value (at generation $g=0$) of the j^{th} parameter of the i^{th} vector is generated by: $x_{j,i,0} = \text{rand}_j[0,1] \cdot (b_{j,U}-b_{j,L}) + b_{j,L}$, $j=1,\dots,D$, $i=1,\dots,Np$

Trial vector generation

At the g^{th} generation, a trial population $P_{u,g}$ consisting of Np D -dimensional trial vectors $\mathbf{v}_{i,g}=[v_{1,i,g},\dots,v_{D,i,g}]$ is generated via mutation and recombination operations applied to the current population $P_{x,g}$

Differential mutation: with respect to each vector $\mathbf{x}_{i,g}$ in the current population, called target vector, a mutant vector $\mathbf{v}_{i,g}$ is generated by adding a scaled, randomly sampled, vector difference to a basis vector randomly selected from the current population

Example: at the g^{th} generation, the i^{th} mutant vector $\mathbf{v}_{i,g}$ with respect to i^{th} target vector $\mathbf{x}_{i,g}$ in the current population is generated by $\mathbf{v}_{i,g} = \mathbf{x}_{r0,g} + F \cdot (\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g})$, $i \neq r0 \neq r1 \neq r2$, mutation scale factor $F \in (0,1+)$

1. EVOLUCIÓN DIFERENCIAL

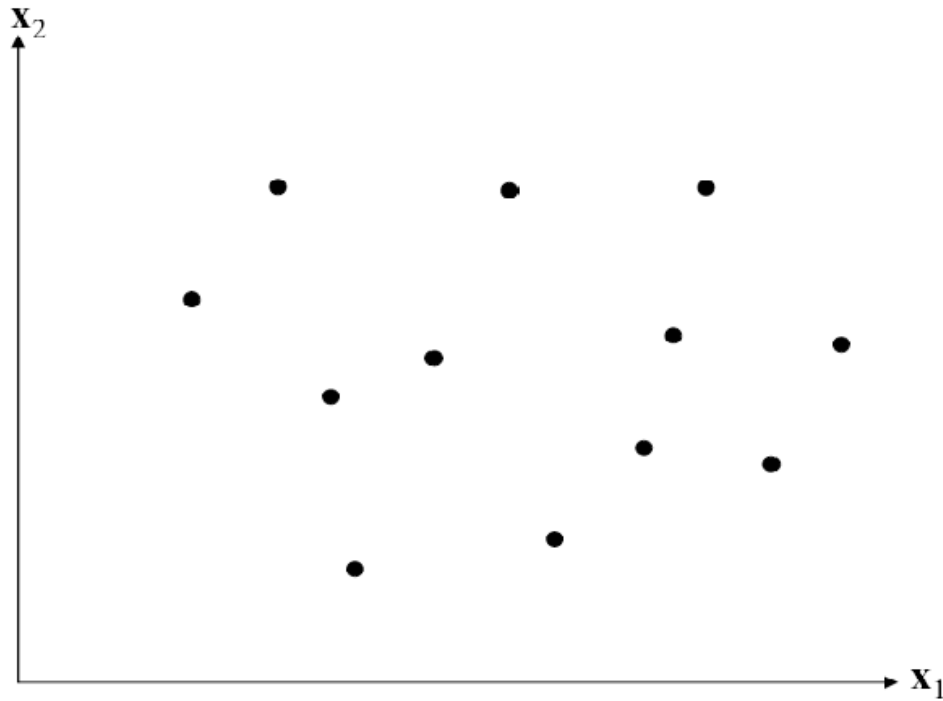


Illustration of classic DE

1. EVOLUCIÓN DIFERENCIAL

Example: at the g^{th} generation, the i^{th} mutant vector $\mathbf{v}_{i,g}$ with respect to i^{th} target vector $\mathbf{x}_{i,g}$ in the current population is generated by $\mathbf{v}_{i,g} = \mathbf{x}_{r0,g} + F \cdot (\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g})$, $i \neq r0 \neq r1 \neq r2$, mutation scale factor $F \in (0, 1+)$

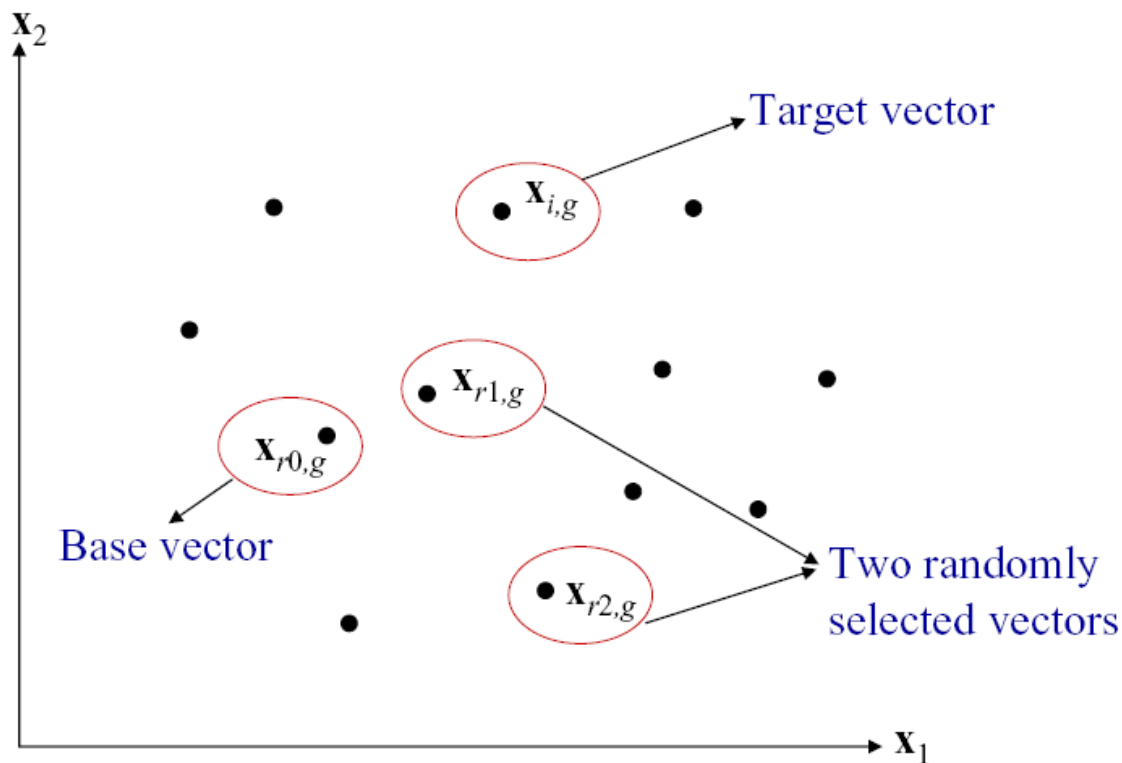
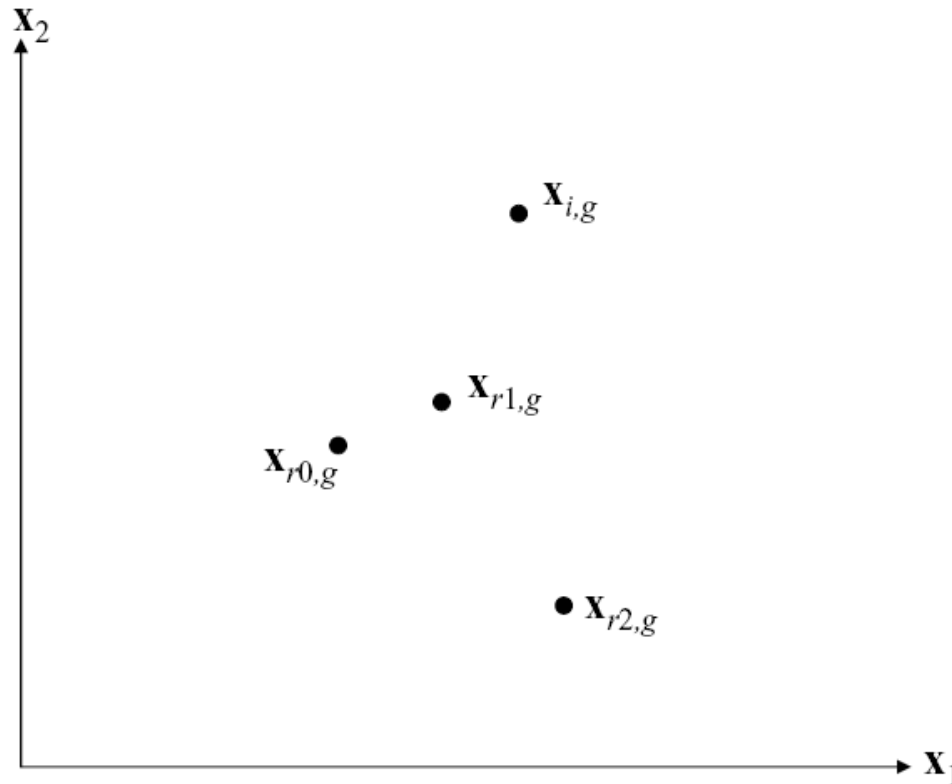


Illustration of classic DE

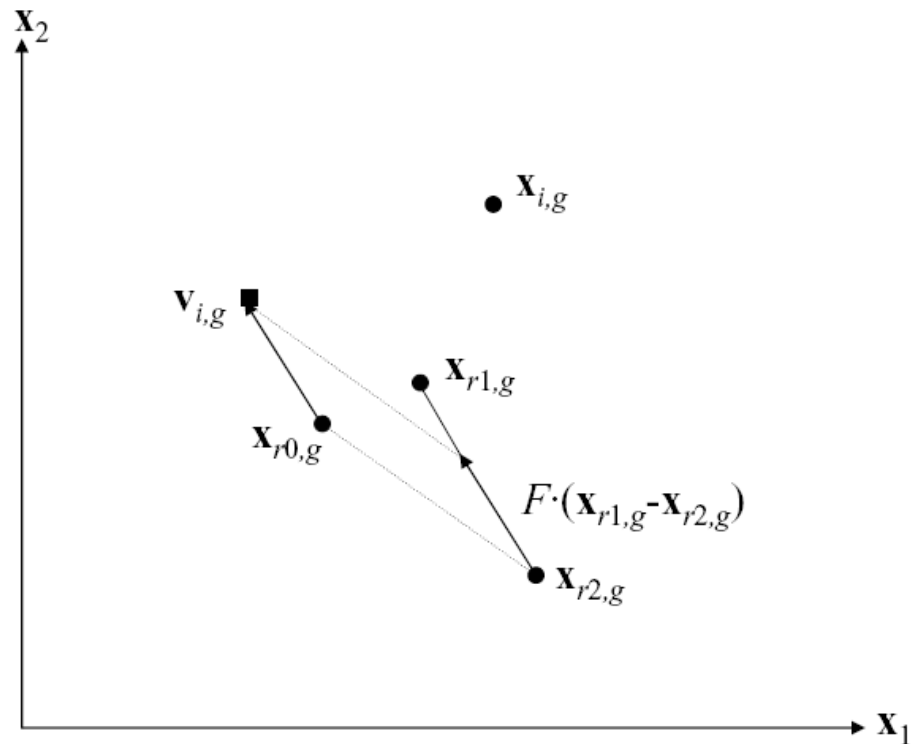
1. EVOLUCIÓN DIFERENCIAL

Example: at the g^{th} generation, the i^{th} mutant vector $\mathbf{v}_{i,g}$ with respect to i^{th} target vector $\mathbf{x}_{i,g}$ in the current population is generated by $\mathbf{v}_{i,g} = \mathbf{x}_{r0,g} + F \cdot (\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g})$, $i \neq r0 \neq r1 \neq r2$, mutation scale factor $F \in (0, 1+)$



1. EVOLUCIÓN DIFERENCIAL

Example: at the g^{th} generation, the i^{th} mutant vector $\mathbf{v}_{i,g}$ with respect to i^{th} target vector $\mathbf{x}_{i,g}$ in the current population is generated by $\mathbf{v}_{i,g} = \mathbf{x}_{r0,g} + F \cdot (\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g})$, $i \neq r0 \neq r1 \neq r2$, mutation scale factor $F \in (0, 1+)$



Mutation

1. EVOLUCIÓN DIFERENCIAL

Discrete recombination: with respect to each target vector $\mathbf{x}_{i,g}$ in the current population, a trial vector $\mathbf{u}_{i,g}$ is generated by crossing the target vector $\mathbf{x}_{i,g}$ with the corresponding mutant vector $\mathbf{v}_{i,g}$ under a pre-specified crossover rate $Cr \in [0,1]$

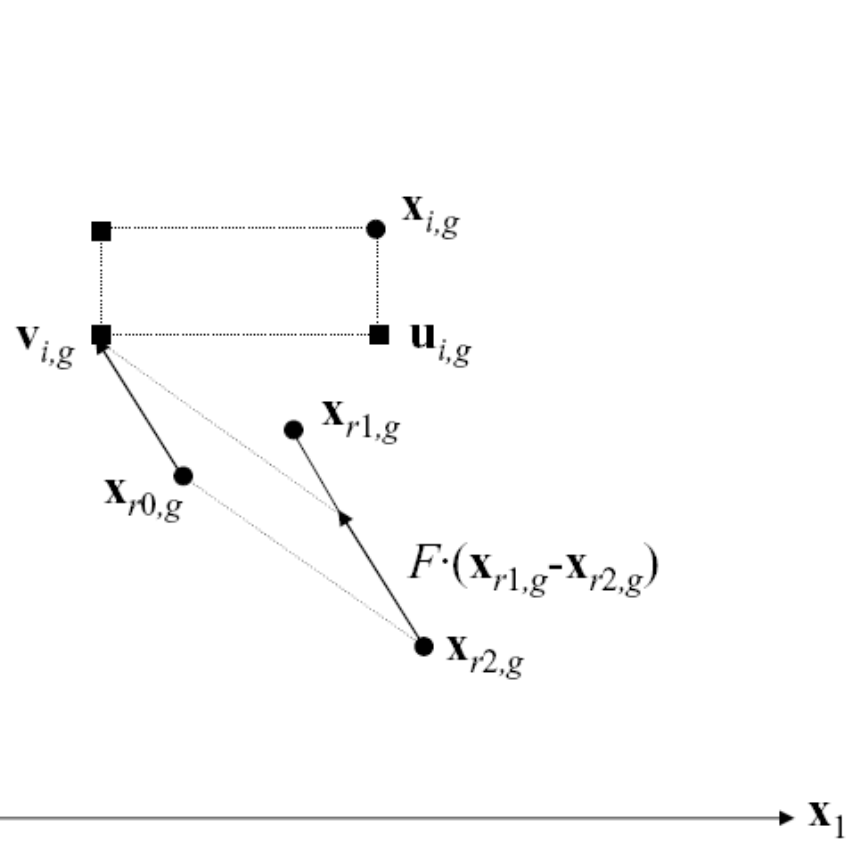
Example: at the g^{th} generation, the i^{th} trial vector $\mathbf{u}_{i,g}$ with respect to i^{th} target vector $\mathbf{x}_{i,g}$ in the current population is generated by:

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

Replacement

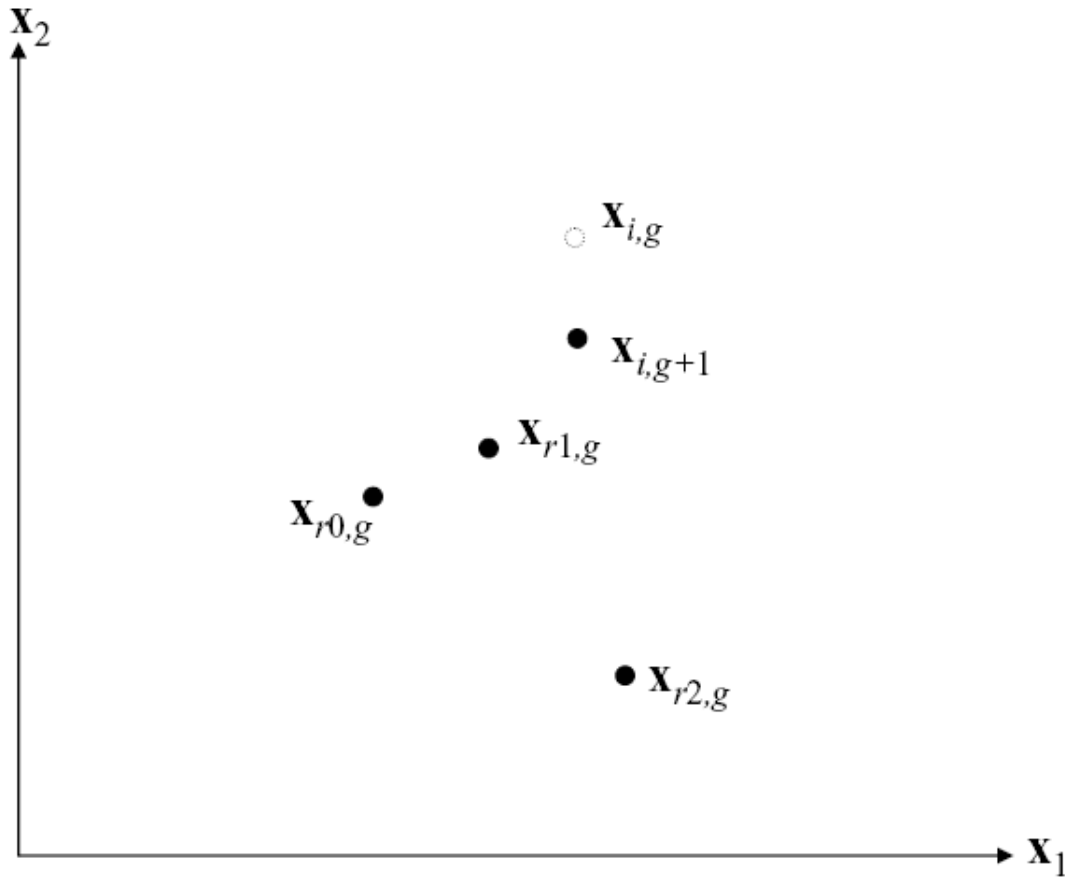
If the trial vector $\mathbf{u}_{i,g}$ has the better objective function value than that of its corresponding target vector $\mathbf{x}_{i,g}$, it replaces the target vector in the $(g+1)^{\text{th}}$ generation; otherwise the target vector remains in the $(g+1)^{\text{th}}$ generation

1. EVOLUCIÓN DIFERENCIAL



Crossover

1. EVOLUCIÓN DIFERENCIAL



Replacement

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 + Y(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} ∪
             {k | Individualk is_better_than Offspringk};
  t = t + 1;}
```

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

2. VARIANTES DE LA EVOLUCIÓN DIFERENCIAL

□ Differential mutation

- One difference vector: $F \cdot (\mathbf{x}_{r1} - \mathbf{x}_{r2})$
- Two difference vector: $F \cdot (\mathbf{x}_{r1} - \mathbf{x}_{r2}) + F \cdot (\mathbf{x}_{r3} - \mathbf{x}_{r4})$
- Mutation scale factor F
 - ❖ Crucial role: balance exploration and exploitation
 - ❖ Dimension dependence: *jitter* (rotation variant) and *dither* (rotation invariant)
 - ❖ Randomization: different distributions of F

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

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

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

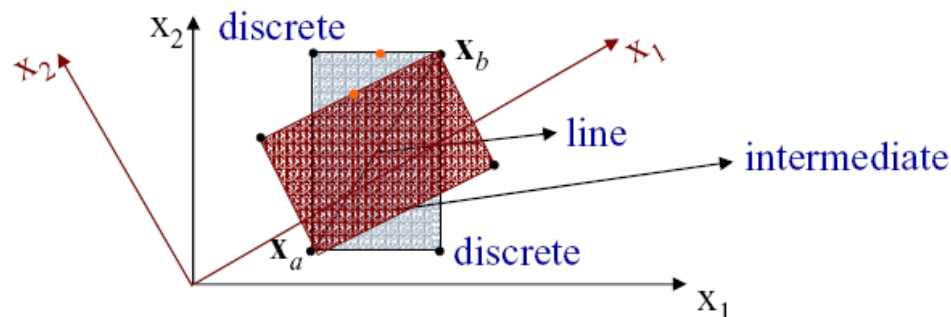
$$\text{DE/rand/2:} \quad \mathbf{V}_{i,G} = \mathbf{X}_{r1,G} + F \cdot (\mathbf{X}_{r2,G} - \mathbf{X}_{r3,G} + \mathbf{X}_{r4,G} - \mathbf{X}_{r5,G})$$

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

2. VARIANTES DE LA EVOLUCIÓN DIFERENCIAL

□ Recombination

- Discrete recombination (crossover) (rotation variant)
 - ❖ One point and multi-point
 - ❖ Exponential
 - ❖ Binominal (uniform)
- Arithmetic recombination
 - ❖ Line recombination (rotation invariant)
 - ❖ Intermediate recombination (rotation variant)
 - ❖ Extended intermediate recombination (rotation variant)



2. VARIANTES DE LA EVOLUCIÓN DIFERENCIAL

➤ Crossover rate $CR \in [0,1]$

Decomposable (small CR) and indecomposable functions (large CR)

❑ Degenerate cases in the trial vector generation

For example, in classic DE, $r_1=r_2$, $r_0=r_1$, $r_0=r_2$, $i=r_0$, $i=r_1$, $i=r_2$

Better to generate mutually exclusive indices for target vector, base vector and vectors constituting the difference vector

Replacement

❑ One-to-one replacement

❑ Neighborhood replacement

2. VARIANTES DE LA EVOLUCIÓN DIFERENCIAL

Practical guideline

$Np \in [5D, 10D]$; Initial choice of $F=0.5$ and $CR=0.1/0.9$;

Increase NP and/or F if premature convergence happens.

2. VARIANTES DE LA EVOLUCIÓN DIFERENCIAL

Motivation for self-adaptation in DE

The performance of DE on different problems depends on:

- Population size
- Strategy and the associated parameter setting to generate trial vectors
- Replacement scheme

It is hard to choose a unique combination to successfully solve any problem at hand

- Population size usually depends on the problem scale and complexity
- During evolution, different strategies coupled with specific parameter settings may favor different search stages
- Replacement schemes influence the population diversity
- Trial and error scheme may be a waste of computational time & resources

Automatically adapt the configuration in DE so as to generate effective trial vectors during evolution

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.



U. K. Chakraborty, *Advances in Differential Evolution, Heidelberg, Germany: Springer-Verlag, 2008.*

3. ALGUNOS TRABAJOS EN EVOLUCIÓN DIFERENCIAL

DE with self-adaptive population [T06]: population size, mutation scale factor and crossover rate are all encoded into each individual.

Self-adaptive DE (2) [BGBMZ06]: encode mutation scale factor and crossover rate in each individual, which are reinitialized according to two new probability variables

[T06] J. Teo, "Exploring dynamic self-adaptive populations in differential evolution," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 10(8): 673-686, 2006.

[BGBMZ06] J. Brest, S. Greiner, B. Boškovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, 10:6 (2006) 646-657

3. ALGUNOS TRABAJOS EN EVOLUCIÓN DIFERENCIAL

Opposition based Differential Evolution

- El algoritmo comprueba constantemente la calidad de una solución y de la de su opuesta en el espacio de búsqueda.
- En teoría, el 50% de las veces, una solución inicial aleatoria está más alejada del óptimo global que su opuesta. Se escoge la mejor de ambas.
- Número Opuesto: $x' = a + b - x$, $x \in [a, b]$
- Durante su ejecución, se ejecuta una fase de salto (condicionada a una probabilidad predefinida) al opuesto de cada individuo y se escogen los mejores. En este caso, no se utilizan los límites del problema, sino que se usan los límites recogidos en la población actual.

3. ALGUNOS TRABAJOS 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: $F \sim N(0.5, 0.3)$, $CR \sim N(0.5, 0.1)$

A.K. Qin, V.L. Huang, P.N. Suganthan. Differential Evolution Algorithm with strategy Adaptation for Global Numerical Optimization. *IEEE Transactions on Evolutionary Computation*, 13:2 (2009) 398-417.

3. ALGUNOS TRABAJOS EN EVOLUCIÓN DIFERENCIAL

DEGL: Differential Evolution with Global and Local Neighborhoods

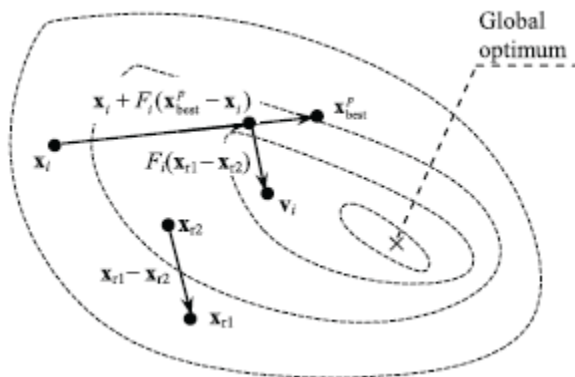
- Utiliza una estrategia DE/target-to-best/1 con alta componente de explotación. Sin embargo, aplica el concepto de vecindad local y vecindad global para determinar el mejor vector, similar a las vecindades en PSO.
- Añade un parámetro de peso (w) que combina el modelo local y global y las vecindades no se establecen acorde a situación geográfica ni fitness, si no a priori en una estructura de anillo.
- Si w es cercano a 1, se favorece la explotación, y si es cercano a 0 se favorece la exploración. Adaptación de w :
 - Incremental
 - Aleatoria
 - AutoAdaptación

S. Das, A. Abraham, U.K. Chakraborty, A. Konar. Differential Evolution Using a Neighborhood-Based Mutation Operator. *IEEE Transactions on Evolutionary Computation*, 13:3 (2009) 526-553.

3. ALGUNOS TRABAJOS EN EVOLUCIÓN DIFERENCIAL

JADE: Adaptive Differential Evolution

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



$$v_{i,g} = x_{i,g} + F_i \cdot (x_{best,g}^p - x_{i,g}) + F_i \cdot (x_{r1,g} - x_{r2,g})$$

- Donde se escoge aleatoriamente uno de los 100p% mejores.
- 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.

J. Zhang, A.C. Sanderson. JADE: Adaptive Differential Evolution with Optional External Archive. *IEEE Transactions on Evolutionary Computation*, 13:5 (2009) 945-958.

3. ALGUNOS TRABAJOS EN EVOLUCIÓN DIFERENCIAL

SFLSDE: Scale Factor Local search Differential Evolution

- Está basado en algoritmos memeticos.
- Utiliza un esquema autoadaptativo con 2 algoritmos de búsqueda local.
- Estos algoritmos de búsqueda local tienen como objetivo detectar valores del factor de escala (F) correspondientes a un descendiente con buen comportamiento.
- Comparan sus resultados con:
 - **J. Brest, S. Greiner, B. Boškovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, 10:6 (2006) 646-657**
 - **S. Rahnamayan, H.R. Tizhoosh, M.M.A. Salama. Opposition-Based Differential Evolution. *IEEE Transactions on Evolutionary Computation*, 12:1 (2008) 64-79.**
 - **N. Noman, H. Iba. Accelerating differential evolution using an adaptive local search. *IEEE Trans Evol Comput* 12:1 (2008)107-125**

F. Neri, V. Tirronen. Scale factor local search in differential evolution. *Memetic Computation* (2009) 1:153-171

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).

La hibridación de los algoritmos de Evolución Diferencial y los algoritmos de búsqueda local son una vía de trabajo con potenciales buenos resultados en el ámbito de la optimización continua.

Review reciente:

Swagatam Das and Ponnuthurai Nagarathnam Suganthan

Differential Evolution: A Survey of the State-of-the-Art,

IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION 15 (1): 4-31 (2011)

BIOINFORMÁTICA

2013 - 2014

PARTE I. INTRODUCCIÓN

- Tema 1. Computación Basada en Modelos Naturales

PARTE II. MODELOS BASADOS EN ADAPTACIÓN SOCIAL (Swarm Intelligence)

- Tema 2. Introducción a los Modelos Basados en Adaptación Social
- Tema 3. Optimización Basada en Colonias de Hormigas
- Tema 4. Optimización Basada en Nubes de Partículas (Particle Swarm)

PARTE III. COMPUTACIÓN EVOLUTIVA

- Tema 5. Introducción a la Computación Evolutiva
- Tema 6. Algoritmos Genéticos I. Conceptos Básicos
- Tema 7. Algoritmos Genéticos II. Diversidad y Convergencia
- Tema 8. Algoritmos Genéticos III. Múltiples Soluciones en Problemas Multimodales
- Tema 9. Estrategias de Evolución y Programación Evolutiva
- Tema 10. Algoritmos Basados en Evolución Diferencial (Differential Evolution – DE)
- **Tema 11. Modelos de Evolución Basados en Estimación de Distribuciones (EDA)**
- Tema 12. Algoritmos Evolutivos para Problemas Multiobjetivo
- Tema 13. Programación Genética
- Tema 14. Modelos Evolutivos de Aprendizaje

PARTE IV. OTROS MODELOS DE COMPUTACIÓN BIOINSPIRADOS

- Tema 15. Sistemas Inmunológicos Artificiales
- Tema 16. Otros Modelos de Computación Natural/Bioinspirados