



**UNIVERSIDAD NACIONAL DE SAN AGUSTÍN**  
**ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN**  
**COMPUTACIÓN BIOINSPIRADA**  
**PRÁCTICA – Fireworks Algorithm**

1. Muestre una iteración del Algoritmo de Fuegos Artificiales para minimizar el siguiente problema:

$$f(x, y) = (x+1) - y$$

$$0 \leq x, y \leq 20$$

Use los siguientes parámetros:

$$n=3, m=3, \hat{m}=2, \hat{A}=10, a=0, b=0.8, X_i=0.01$$

Muestre las ubicaciones de los fuegos artificiales, el fitness de cada ubicación, el número de chispas y la amplitud para cada fuego artificial, las ubicaciones de las chispas generadas por los dos tipos de explosiones, las probabilidades de las ubicaciones de los fuegos artificiales y las chispas generadas, y la mejor ubicación al final de la iteración.

---

**Algorithm 3.** Framework of the FA

---

Randomly select  $n$  locations for fireworks;

**while** stop criteria=false **do**

    Set off  $n$  fireworks respectively at the  $n$  locations;

**for** each firework  $x_i$  **do**

        Calculate the number of sparks that the firework yields:  $\hat{s}_i$ ;

        Obtain locations of  $\hat{s}_i$  sparks of the firework  $x_i$ ;

**end for**

**for**  $k=1:\hat{m}$  **do**

        Randomly select a firework  $x_j$ ;

        Generate a specific spark for the firework;

**end for**

    Select the best location and keep it for next explosion generation;

    Randomly select  $n - 1$  locations from the two types of sparks and the current fireworks according to the probability;

**end while**

---

## **PRACTICA FIREFLY ALGORITHM**

1. Muestre una iteración del Firefly Algorithm para maximizar el siguiente problema:

$$f(x,y) = -x^2 - y^2$$

$$\begin{aligned} -5 \leq x \leq 5 \\ -5 \leq y \leq 5 \end{aligned}$$

Considere los siguientes parámetros:

$$\alpha = 0.2$$

$$S_k = 1$$

$$\beta_0 = 1$$

$$\gamma = 1$$

Muestre:

- La distancia entre abejas
- El numero *random*
- Las nuevas posiciones y brillo



**UNIVERSIDAD NACIONAL DE SAN AGUSTÍN**  
**ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN**  
**COMPUTACIÓN BIOINSPIRADA**  
**PRÁCTICA – PLANT PROPAGATION ALGORITHM**

1. Muestre una iteración del Plant Propagation Algorithm para minimizar el siguiente problema.

$$f(x,y) = (x-2)^2 + (y-1)^2$$

$$-5 \leq x, y \leq 5$$

Considere los siguientes parámetros:

- Tamaño de la población: 4
- Número de generaciones(g\_max):2
- Número máximo de corredores por solución(n\_max): 5

Muestre las soluciones con su valor de la función objetivo  $f(x)$  y el valor de mapeamiento N. Además, muestre el numero de corredores para cada solución , su distancia y la nueva soluciones generada.



UNIVERSIDAD NACIONAL DE SAN AGUSTÍN  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN  
COMPUTACIÓN BIOINSPIRADA  
CAPÍTULO II: COMPUTACIÓN SOCIAL  
LABORATORIO 13

**I. TEMA: ABC**

**II. DOCENTE:** Dr. Edward Hinojosa Cárdenas

**III. FECHA:** 30 de mayo del 2019

**IV. PROPÓSITO**

1. Aplicar el Algoritmo de Colonia de Abejas para maximizar la siguiente función:

$$f(x, y) = xsin(4\pi x) - ysin(4\pi y + \pi) + 1, \quad x, y \in [-1, 2]$$

- Pruebe con diferentes valores en los parámetros.
- Muestre los valores obtenidos como en los ejemplos vistos en teoría.

**V. CONCEPTOS BÁSICOS**

1. Algoritmo

```
1 Begin
2   Inicializar la población de soluciones  $\mathbf{x}_{i,0}$ ,  $i = 1, \dots, SN$ 
3   Evaluar la población
4    $g = 1$ 
5   Repeat
6     Producir nuevas soluciones  $\mathbf{v}_{i,g}$  para las abejas empleadas
      y evaluarlas
7     Conservar la mejor solución entre la actual y la candidata
8     Seleccionar las soluciones que serán visitadas por una abeja
      observadora según su aptitud
9     Producir nuevas soluciones  $\mathbf{v}_{i,g}$  para las abejas observadoras
      y evaluarlas
10    Conservar la mejor solución entre la actual y la candidata
11    Determinar si existe una fuente abandonada y reemplazarla
      utilizando una abeja exploradora
12    Memorizar la mejor solución encontrada hasta este momento
13     $g = g + 1$ 
14 Until  $g = MCN$ 
15 End
```



## **VI. EQUIPOS Y MATERIALES**

1. Computador

## **VII. NOTAS DE SEGURIDAD**

Usar la computadora y los servicios de la universidad adecuadamente con las indicaciones del docente.

## **VIII. CONCLUSIONES**

Al finalizar el estudiante deberá:

1. Presentar al profesor el resultado de su implementación.
2. Generar un archivo .txt con el resultado obtenido en ambos ejercicios.
3. Compactar el código junto en una carpeta, más el resultado obtenido y subir el archivo compactado al aula virtual (teniendo del día martes 04/06 hasta las 23:55pm).



UNIVERSIDAD NACIONAL DE SAN AGUSTÍN  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN  
COMPUTACIÓN BIOINSPIRADA  
**CAPÍTULO III: SISTEMAS INMUNOLÓGICOS ARTIFICIALES**  
**LABORATORIO 14**

**I. TEMA: ALGORITMO DE SELECCIÓN CLONAL**

**II. DOCENTE:** Dr. Edward Hinojosa Cárdenas

**III. FECHA:** 13 de junio del 2019

**IV. PROPÓSITO**

1. Aplicar el Algoritmo de Selección Clonal para minimizar la siguiente función:

$$f(x_1, x_2) = -\cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$$

$$-10 \leq x_1 \leq 10, -10 \leq x_2 \leq 10$$

- Pruebe con diferentes valores en los parámetros.
- Muestre los valores obtenidos como en los ejemplos vistos en teoría.

**V. CONCEPTOS BÁSICOS**

1. Algoritmo

```
Input: Population_size, Selection_size, Problem_size, RandomCells_num, Clone_rate, Mutation_rate
Output: Population
Population ← CreateRandomCells(Population_size, Problem_size)
While (→StopCondition())
    For (pi ∈ Population)
        Affinity(pi)
    End
    Population_select ← Select(Population, Selection_size)
    Population_clones ← ∅
    For (pi ∈ Population_select)
        Population_clones ← clone(pi, Clone_rate)
    End
    For (pi ∈ Population_clones)
        Hypermutate(pi, Mutation_rate)
        Affinity(pi)
    End
    Population ← Select(Population, Population_clones, Population_size)
    Population_rand ← CreateRandomCells(RandomCells_num)
    Replace(Population, Population_rand)
End
Return (Population)
```

Pseudocode for CLONALG.



## **VI. EQUIPOS Y MATERIALES**

1. Computador

## **VII. NOTAS DE SEGURIDAD**

Usar la computadora y los servicios de la universidad adecuadamente con las indicaciones del docente.

## **VIII. CONCLUSIONES**

Al finalizar el estudiante deberá:

1. Presentar al profesor el resultado de su implementación.
2. Generar un archivo .txt con el resultado obtenido.
3. Compactar el código junto en una carpeta, más el resultado obtenido y subir el archivo compactado al aula virtual (teniendo del día jueves 13/06 hasta las 23:55pm).



UNIVERSIDAD NACIONAL DE SAN AGUSTÍN  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN  
COMPUTACIÓN BIOINSPIRADA  
**CAPÍTULO III: SISTEMAS INMUNOLÓGICOS ARTIFICIALES**  
**LABORATORIO 15**

**I. TEMA: ALGORITMO DE RED INMUNE**

**II. DOCENTE:** Dr. Edward Hinojosa Cárdenas

**III. FECHA:** 13 de junio del 2019

**IV. PROPÓSITO**

1. Aplicar el Algoritmo de Red Inmune para optimización para minimizar la siguiente función:

$$f(x_1, x_2) = -\cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$$

$$-10 \leq x_1 \leq 10, -10 \leq x_2 \leq 10$$

- Pruebe con diferentes valores en los parámetros.
- Muestre los valores obtenidos como en los ejemplos vistos en teoría.

**V. CONCEPTOS BÁSICOS**

1. Algoritmo

```
Input: Populationsize, ProblemSize, Nclones, Nrandom, AffinityThreshold
Output: Sbest
Population ← InitializePopulation(Populationsize, ProblemSize)
While ( $\neg$ StopCondition())
    EvaluatePopulation(Population)
    Sbest ← GetBestSolution(Population)
    Progeny ←  $\emptyset$ 
    Costavg ← CalculateAveragePopulationCost(Population)
    While (CalculateAveragePopulationCost(Population) > Costavg)
        For Celli ∈ Population
            Clones ← CreateClones(Celli, Nclones)
            For Clonei ∈ Clones
                Clonei ← MutateRelativeToFitnessOfParent(Clonei, Celli)
            End
            EvaluatePopulation(Clones)
            Progeny ← GetBestSolution(Clones)
        End
        SuppressLowAffinityCells(Progeny, AffinityThreshold)
        Progeny ← CreateRandomCells(Nrandom)
        Population ← Progeny
    End
Return (Sbest)
```

Pseudocode for opt-aiNet.



## **VI. EQUIPOS Y MATERIALES**

1. Computador

## **VII. NOTAS DE SEGURIDAD**

Usar la computadora y los servicios de la universidad adecuadamente con las indicaciones del docente.

## **VIII. CONCLUSIONES**

Al finalizar el estudiante deberá:

1. Presentar al profesor el resultado de su implementación.
2. Generar un archivo .txt con el resultado obtenido.
3. Compactar el código junto en una carpeta, más el resultado obtenido y subir el archivo compactado al aula virtual (teniendo del día martes 18/06 hasta las 23:55pm).



UNIVERSIDAD NACIONAL DE SAN AGUSTÍN  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN  
COMPUTACIÓN BIOINSPIRADA  
**CAPÍTULO III: SISTEMAS INMUNOLÓGICOS ARTIFICIALES**  
**LABORATORIO 16**

**I. TEMA: BACTERIAL FORAGING OPTIMIZATION ALGORITHM**

**II. DOCENTE:** Dr. Edward Hinojosa Cárdenas

**III. FECHA:** 13 de junio del 2019

**IV. PROPÓSITO**

1. Aplicar el Bacterial Foraging Optimization Algorithm para minimizar la siguiente función:

$$f(x_1, x_2) = -\cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$$

$$-10 \leq x_1 \leq 10, -10 \leq x_2 \leq 10$$

- Pruebe con diferentes valores en los parámetros.
- Muestre los valores obtenidos como en los ejemplos vistos en teoría.

**V. CONCEPTOS BÁSICOS**

1. Algoritmo

```
Input: Problem_size, Cells_num, N_ed, N_re, N_c, N_s, Step_size, d_attract, w_attract, h_repellant, w_repellant, P_ed
Output: Cell_best
Population ← InitializePopulation(Cells_num, Problem_size)
For (l = 0 To N_ed)
    For (k = 0 To N_re)
        For (j = 0 To N_c)
            ChemotaxisAndSwim(Population, Problem_size, Cells_num, N_s, Step_size, d_attract, w_attract, h_repellant,
w_repellant)
            For (Cell ∈ Population)
                If (Cost(Cell) ≤ Cost(Cell_best))
                    Cell_best ← Cell
            End
        End
    End
    SortByCellHealth(Population)
    Selected ← SelectByCellHealth(Population,  $\frac{Cells\_num}{2}$ )
    Population ← Selected
    Population ← Selected
End
For (Cell ∈ Population)
    If (Rand() ≤ P_ed)
        Cell ← CreateCellAtRandomLocation()
    End
End
End
Return (Cell_best)
```

Pseudocode for the BFOA.



## **VI. EQUIPOS Y MATERIALES**

1. Computador

## **VII. NOTAS DE SEGURIDAD**

Usar la computadora y los servicios de la universidad adecuadamente con las indicaciones del docente.

## **VIII. CONCLUSIONES**

Al finalizar el estudiante deberá:

1. Presentar al profesor el resultado de su implementación.
2. Generar un archivo .txt con el resultado obtenido.
3. Compactar el código junto en una carpeta, más el resultado obtenido y subir el archivo compactado al aula virtual (teniendo del día martes 09/07 hasta las 23:55pm).



UNIVERSIDAD NACIONAL DE SAN AGUSTÍN  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN  
COMPUTACIÓN BIOINSPIRADA  
**CAPÍTULO III: SISTEMAS INMUNOLÓGICOS ARTIFICIALES**  
**LABORATORIO 17**

**I. TEMA: BACTERIAL EVOLUTIONARY ALGORITHM**

**II. DOCENTE:** Dr. Edward Hinojosa Cárdenas

**III. FECHA:** 13 de junio del 2019

**IV. PROPÓSITO**

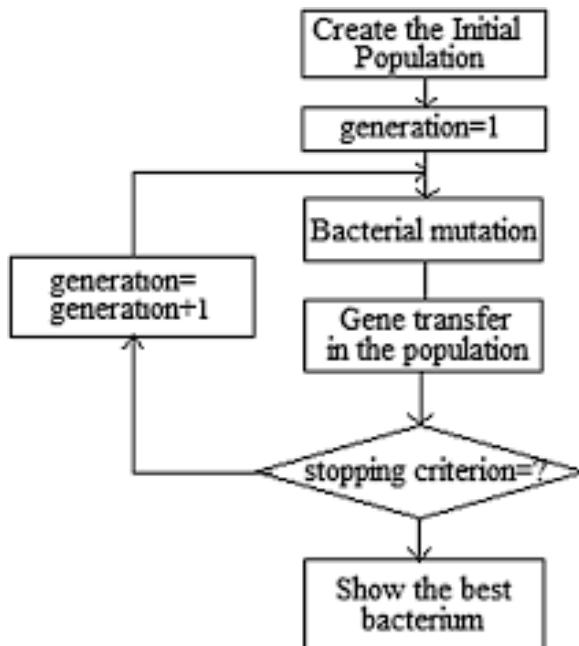
1. Aplicar el Bacterial Evolutionary Algorithm para generar un Sistema de Clasificación Basado en Reglas Fuzzy para la base de datos iris.

<https://archive.ics.uci.edu/ml/datasets/iris>

- Pruebe con diferentes valores en los parámetros.
- Muestre los valores obtenidos como en los ejemplos vistos en teoría.

**V. CONCEPTOS BÁSICOS**

1. Algoritmo





## **VI. EQUIPOS Y MATERIALES**

1. Computador

## **VII. NOTAS DE SEGURIDAD**

Usar la computadora y los servicios de la universidad adecuadamente con las indicaciones del docente.

## **VIII. CONCLUSIONES**

Al finalizar el estudiante deberá:

1. Presentar al profesor el resultado de su implementación.
2. Generar un archivo .txt con el resultado obtenido.
3. Compactar el código junto en una carpeta, más el resultado obtenido y subir el archivo compactado al aula virtual (teniendo del día martes 09/07 hasta las 23:55pm).