# The Whale Optimization Algorithm (WOA)

Ignacio Aguilera Martos

22 Junio 2018

Metaheurísticas

# Contenidos

# Introducción del problema

# Competición CEC2014

# Competición CEC2014

## WOA

- Algoritmo bioinspirado en cómo cazan las ballenas jorobadas.
- Hecho por Seyedali Mirjalili y Andrew Lewis.
- Se ejecuta el algoritmo sobre las 20 primeras funciones de CEC2014.

## CEC2014

- Es una competición reconocida a nivel mundial.
- Se intenta resolver un problema de minimizacion con 30 funciones.
- En nuestro caso sólo lo hemos hecho con dimensión 10 y 30 aunque en la competición se hacía con dimensiones 10,30,50 y 100.
- El ganador de la competición fue L-SHADE.

# Descripción del algoritmo inicial

# Fases de la caza

# Fases de la caza

## Fases de la caza

- Exploración para encontrar presas.

# Fases de la caza

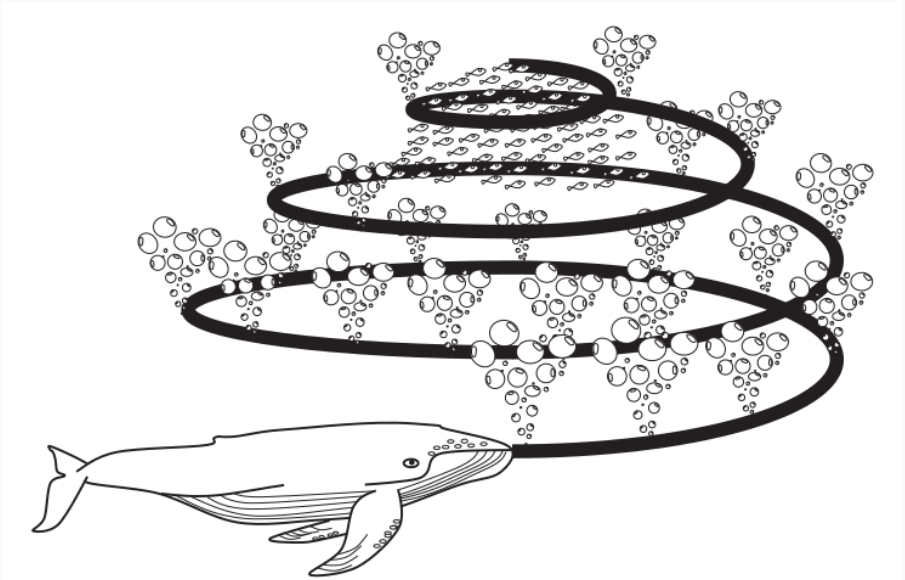
## Fases de la caza

- Exploración para encontrar presas.
- Caza de presas.

# Fases de la caza

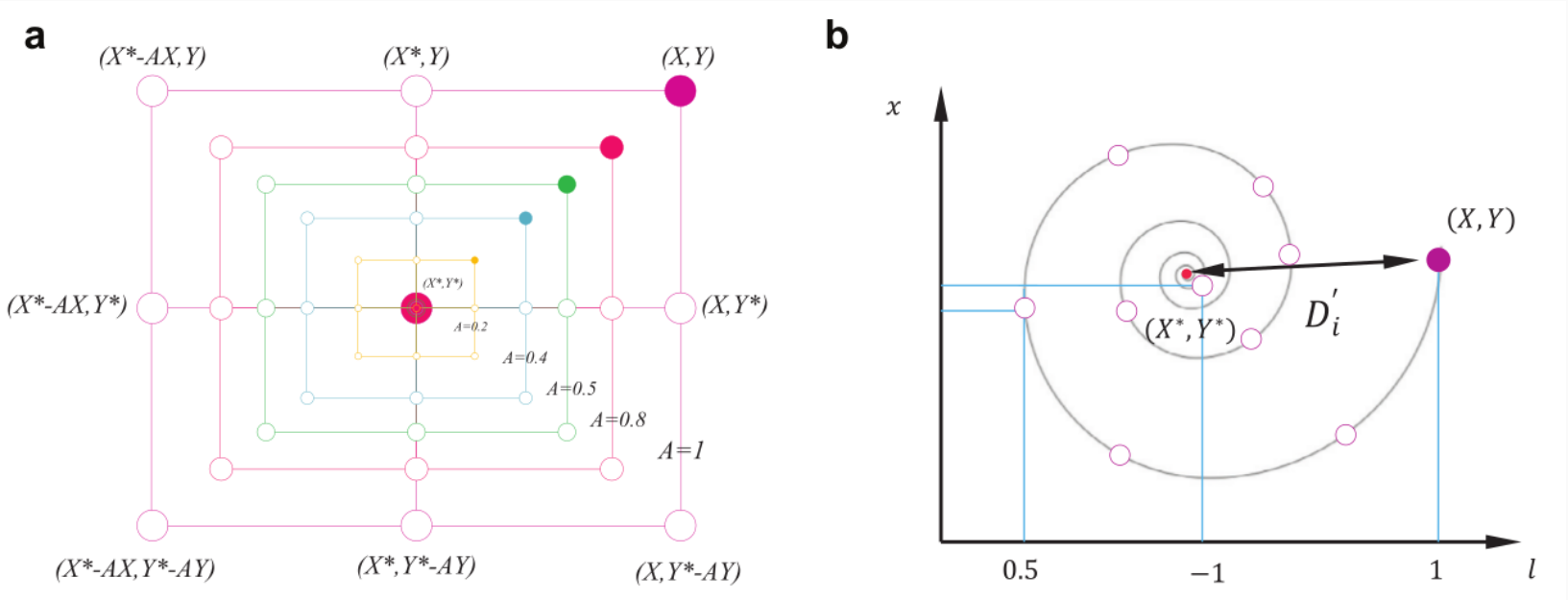
## Fases de la caza

- Exploración para encontrar presas.
- Caza de presas.

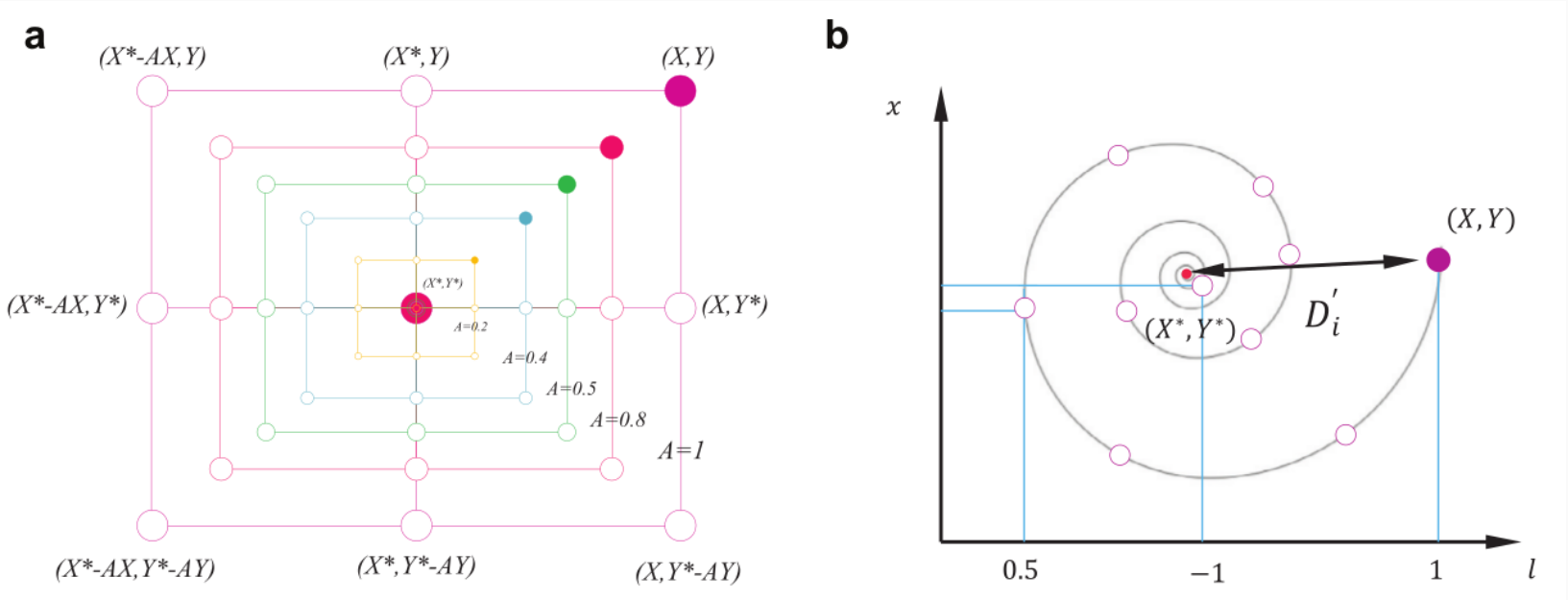


# Modelo matemático

# Aproximación a la presa

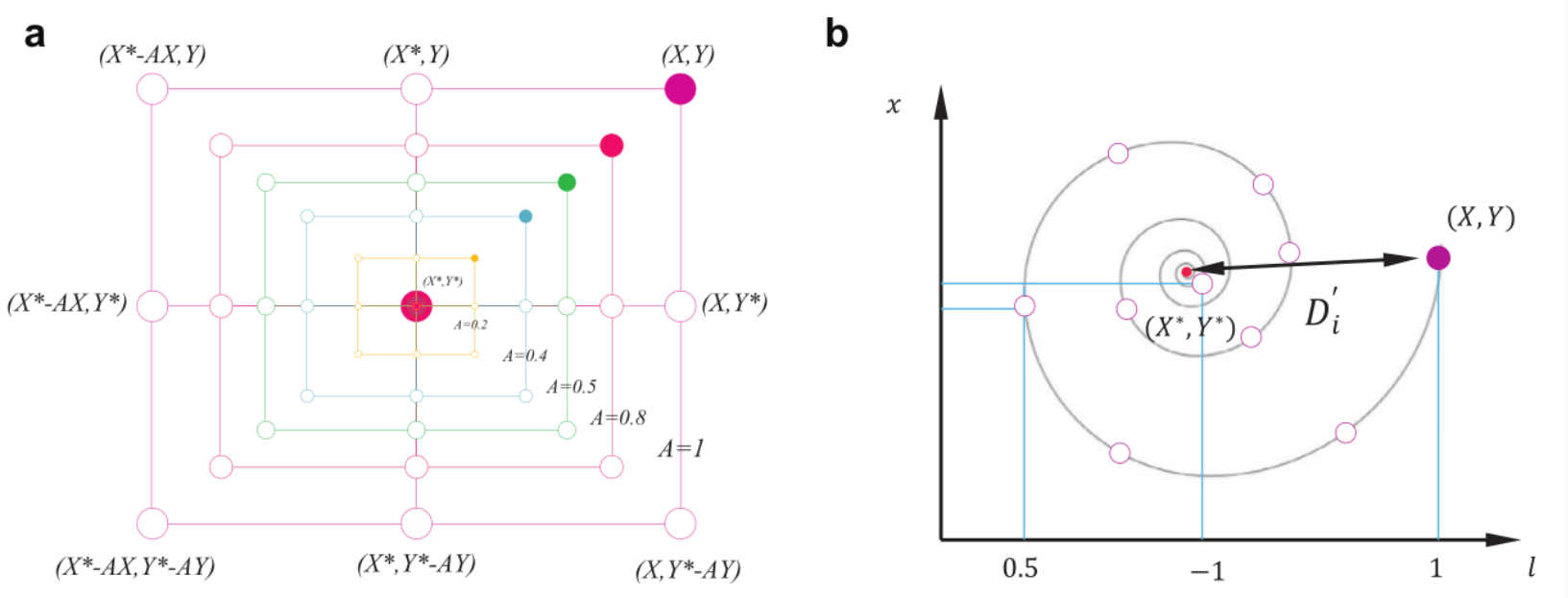


# Aproximación a la presa




$$D(t) = |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)| \qquad \vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot D(t)$$

# Aproximación a la presa

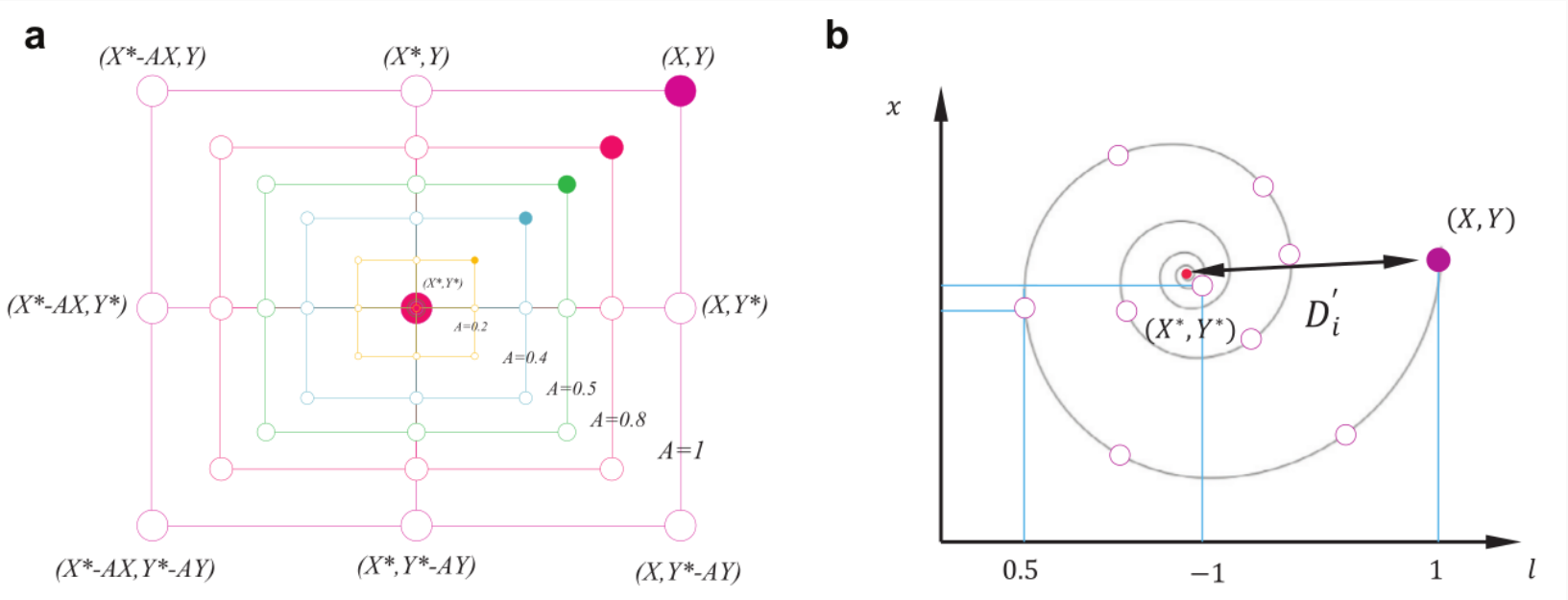



$$D(t) = |\vec{C} \cdot \vec{X^*}(t) - \vec{X}(t)| \qquad \vec{X}(t+1) = \vec{X^*}(t) - \vec{A} \cdot D(t)$$

$$D'(t) = |\vec{X^*}(t) - \vec{X}(t)| \qquad \vec{X}(t+1) = \vec{D'}(t) \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X^*}(t)$$

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r} - \vec{a} \qquad \vec{C} = 2 \cdot \vec{r}$$

# Aproximación a la presa




$$D(t) = |\vec{C} \cdot \vec{X^*}(t) - \vec{X}(t)| \qquad \vec{X}(t+1) = \vec{X^*}(t) - \vec{A} \cdot D(t)$$

$$D'(t) = |\vec{X^*}(t) - \vec{X}(t)| \qquad \vec{X}(t+1) = \vec{D'}(t) \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X^*}(t)$$

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r} - \vec{a} \qquad \vec{C} = 2 \cdot \vec{r}$$

Donde $X$ es la posición de la ballena, $X^*$ la posición de la presa, $\vec{r}$ un vector aleatorio con valores en el intervalo $[0, 1]$ y $a \in [0, 2]$ que se decrementa de forma lineal desde 2 hasta 0.

# Ecuación real del movimiento

# Ecuación real del movimiento

$$\vec{X}(t+1) = \begin{cases} \vec{X}(t+1) = \vec{X^*}(t) - \vec{A} \cdot D(t) & si\ \ p < 0{,}5 \\ \vec{X}(t+1) = \vec{D'}(t) \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X^*}(t) & si\ \ p \geq 0{,}5 \end{cases}$$

# Ecuación real del movimiento

$$\vec{X}(t+1) = \begin{cases} \vec{X}(t+1) = \vec{X^*}(t) - \vec{A} \cdot D(t) & si\ p < 0{,}5 \\ \vec{X}(t+1) = \vec{D'}(t) \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X^*}(t) & si\ p \geq 0{,}5 \end{cases}$$

Donde p es un número aleatorio en el intervalo $[0, 1]$

# Ecuación real del movimiento

$$\vec{X}(t+1) = \begin{cases} \vec{X}(t+1) = \vec{X^*}(t) - \vec{A} \cdot D(t) & si\ p < 0{,}5 \\ \vec{X}(t+1) = \vec{D'}(t) \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X^*}(t) & si\ p \geq 0{,}5 \end{cases}$$

Donde p es un número aleatorio en el intervalo $[0, 1]$
En caso de no tener presa hacemos el movimiento lineal hacia una ballena aleatoria.

# Pseudocódigo

*Initialize the whales population* $X_i$ *(i = 1, 2, ..., n)*
*Calculate the fitness of each search agent*
*X*=the best search agent*
***while*** *(t < maximum number of iterations)*
    ***for*** *each search agent*
    *Update a, A, C, l, and p*
        ***if1*** *(p<0.5)*
            ***if2*** *(|A| < 1)*
                *Update the position of the current search agent by the Eq. (2.1)*
            ***else if2*** *(|A| ≥ 1)*
                *Select a random search agent* ($X_{rand}$)
                *Update the position of the current search agent by the Eq. (2.8)*
            ***end if2***
        ***else if1*** *(p≥0.5)*
                *Update the position of the current search by the Eq. (2.5)*
        ***end if1***
    ***end for***
    *Check if any search agent goes beyond the search space and amend it*
    *Calculate the fitness of each search agent*
    *Update X* if there is a better solution*
    *t=t+1*
***end while***
*return X**

# Desarrollo de mejoras

# Desarrollo de mejoras

# Desarrollo de mejoras

## Fases

- Inicialización aleatoria de las ballenas.
- Hacer una aproximación espiral a una solución aleatoria al principio del algoritmo.
- Incorporación de la búsqueda local Solis Wets.
- Incorporación de un esquema de Differential Evolution.
- Sustitución de Solis Wets por CMAES.
- Reemplazar la aproximación aleatoria para aproximarse a una posición aleatoria del espacio en vez de a una ballena aleatoria.

# Versión final

# Versión final

**Algorithm 1** Ballena(f_obj,inf,sup,dimension,nBallenas)

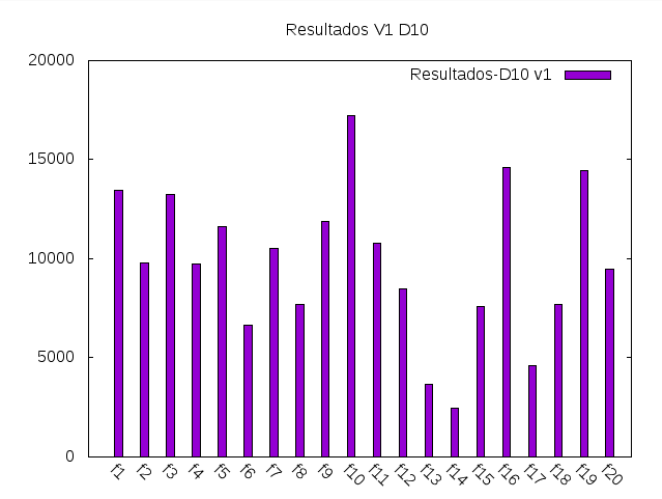
max_evals = $10000 \cdot dimension$
evaluaciones = 0
Inicializo lider_pos a un vector aleatorio con valores entre inf y sup.
lider_score = $\infty$
Genero una población inicial llamada posiciones con vectores aleatorios con valores entre inf y sup.
t = 0
max_iter = (0.9*max_evals)/nBallenas
a = 2
Coloco en un principio el fitness de cada ballena como infinito en el vector fitness.
**while** *evaluaciones* < *max_evals* **do**
  **if** $t\,\%100 == 0$ and $t! = 0$ **then**
    Tomamos un 25 % de las ballenas de forma aleatoria.
    Ejecutamos CMAES sobre el 25 % elegido
    Actualizamos el vector de posiciones con las soluciones de CMAES.
    Actualizamos el vector fitness con el fitness de las soluciones.
    Sumamos a evaluaciones las evaluaciones consumidas por CMAES.
  **end if**
  **if** $t\,\%50 == 0$ and $t! = 0$ **then**
    Aplico el esquema de Differential Evolution sobre el vector posiciones.
    Actualizo el vector posiciones, el vector fitness y sumo las evaluaciones consumidas por Differential Evolution.
  **end if**
  Comprobamos si las ballenas se han salido de los límites, actualizamos sus fitness y actualizamos lider_pos y lider_score si es necesario.
  Sumamos nBallenas a evaluaciones.

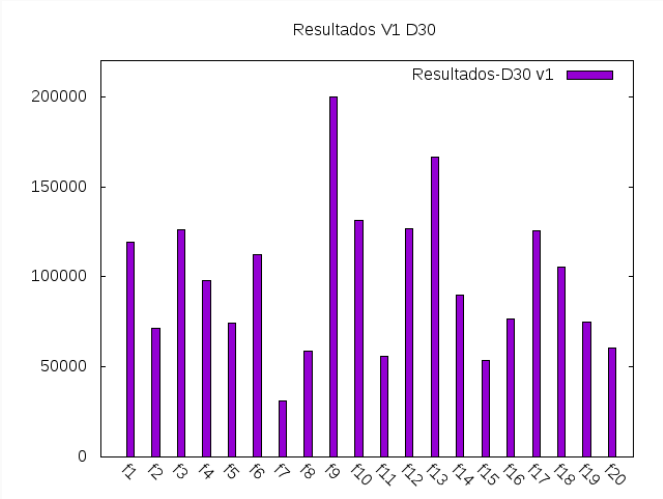
# Versión final

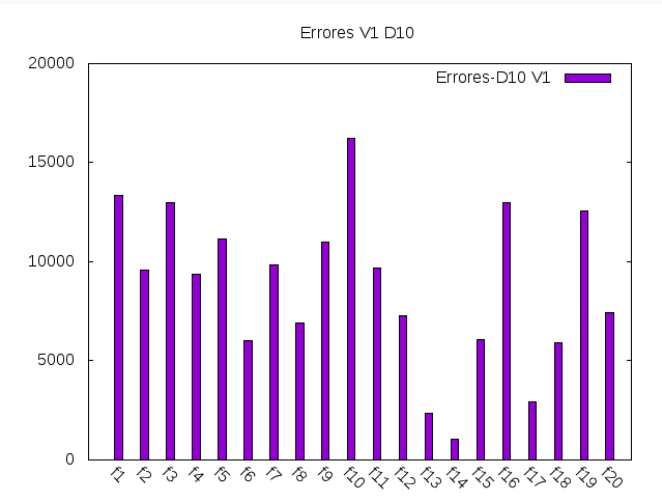
$a = 2 - t \cdot \frac{2}{max\_iter}$
**for** i=0,...,nBallenas-1 **do**
Tomamos dos números aleatorios entre 0 y 1 r1 y r2.
$A = 2 \cdot a \cdot r1 - a$
$C = 2 \cdot r2$
Se toma p un número aleatorio entre 0 y 1.
**if** $p < 0{,}9$ **then**
**if** $|A| \geq 1$ **then**
Tomamos X_rand un vector aleatorio con valores en el intervalo $[inf, sup]$
$D\_X\_rand = |C \cdot X\_rand - posiciones[i]|$
$posiciones[i] = X\_rand - A \cdot D\_X\_rand$
**else**
$D\_lider = |C \cdot lider\_pos - posiciones[i]|$
$posiciones[i] = lider\_pos - A \cdot D\_lider$
**end if**
**else**
Tomamos la mitad peor de las ballenas y las reemplazamos por vectores aleatorios.
**end if**
**end for**
**end while**
Comprobamos si las ballenas se han salido de los límites, actualizamos sus fitness y actualizamos lider_pos y lider_score si es necesario.
Aplicamos CMAES sobre lider_pos.
**return** lider_pos, lider_score

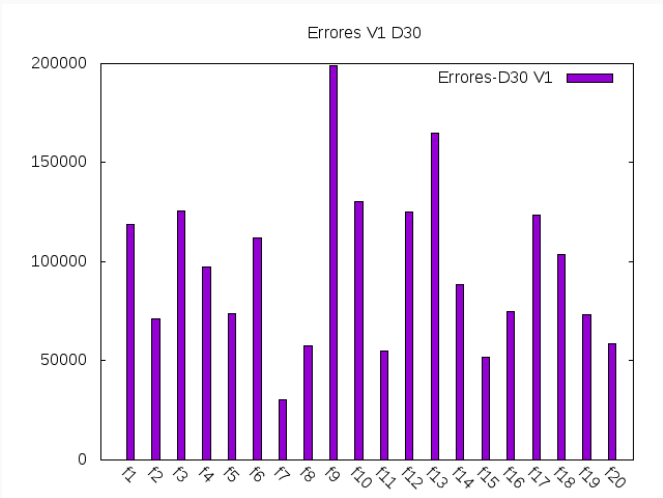
# Resultados

# Versión inicial

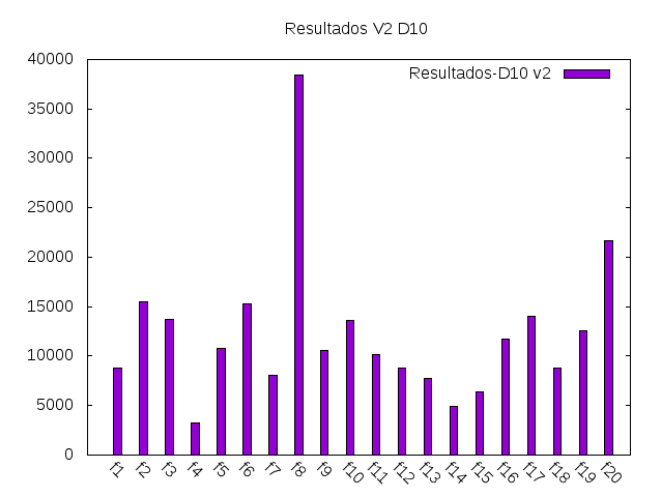


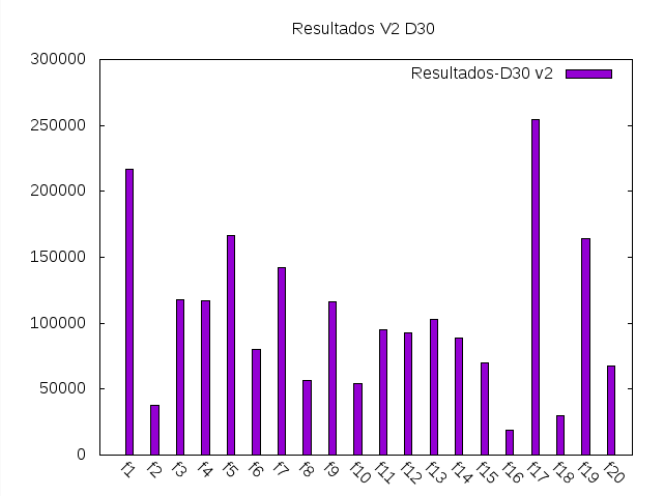



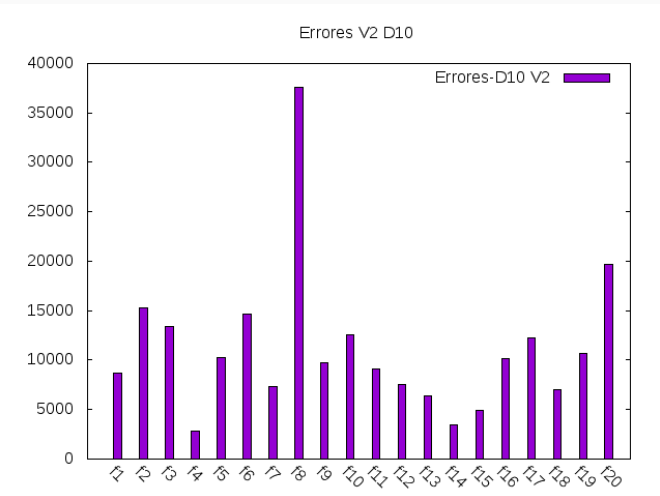



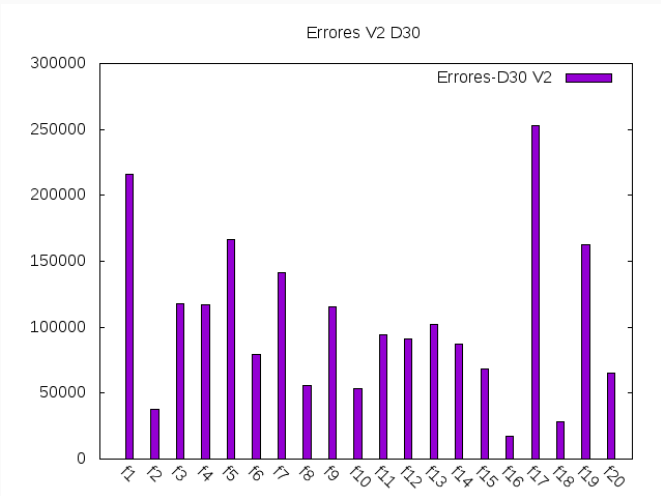



# Segunda versión



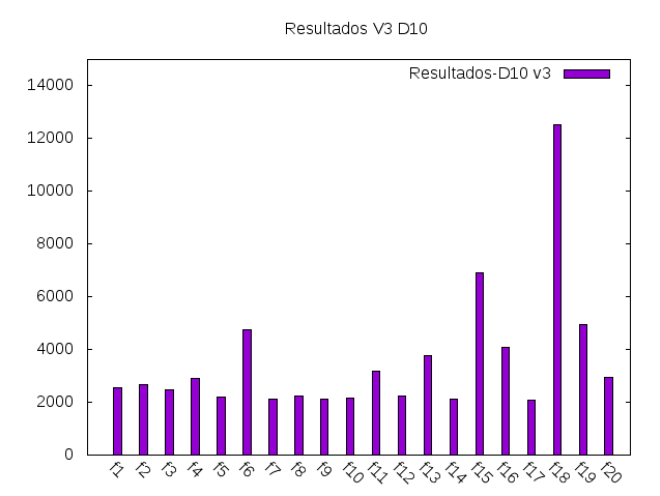



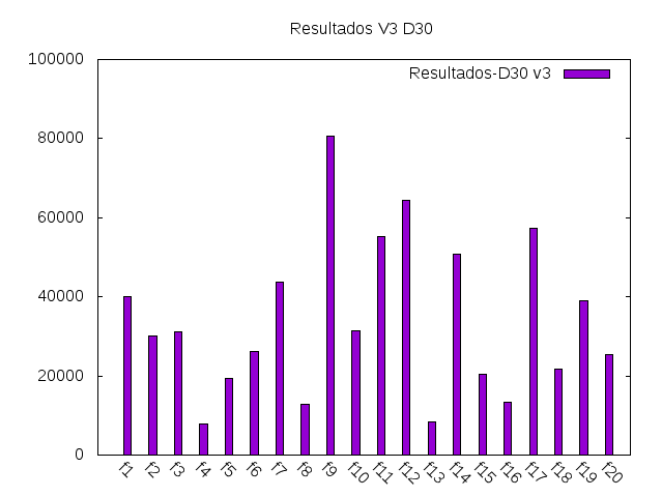



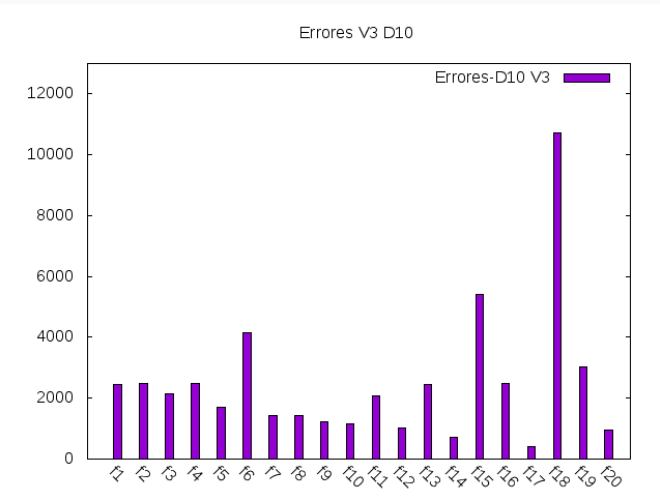

## Características

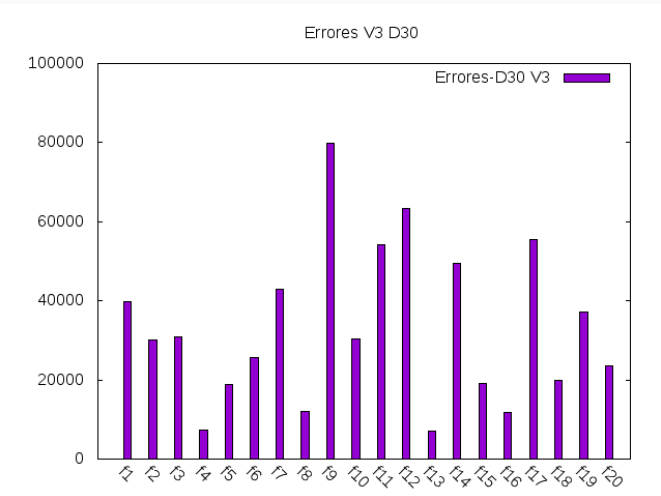
- Hace una aproximación en espiral a una solución aleatoria.

# Tercera versión



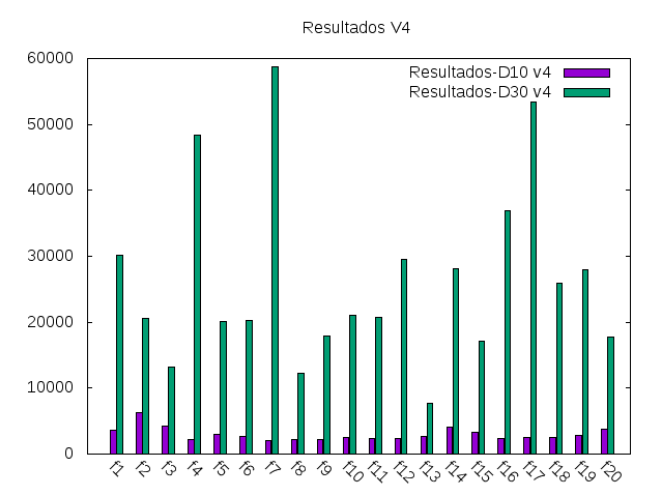



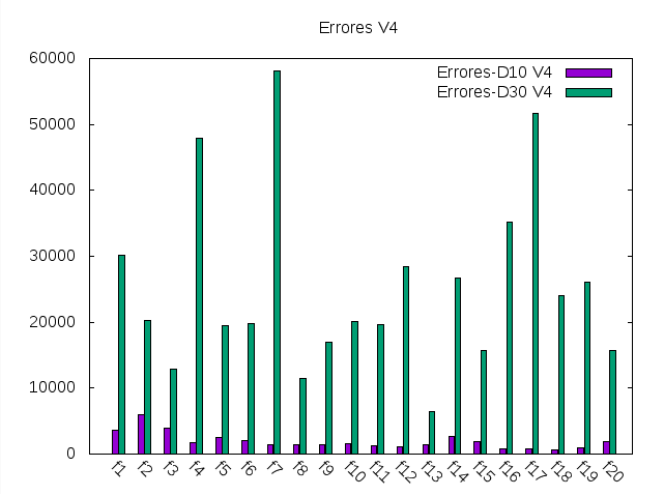




## Características

- Incorporación de la búsqueda local Solis Wets.

# Cuarta versión

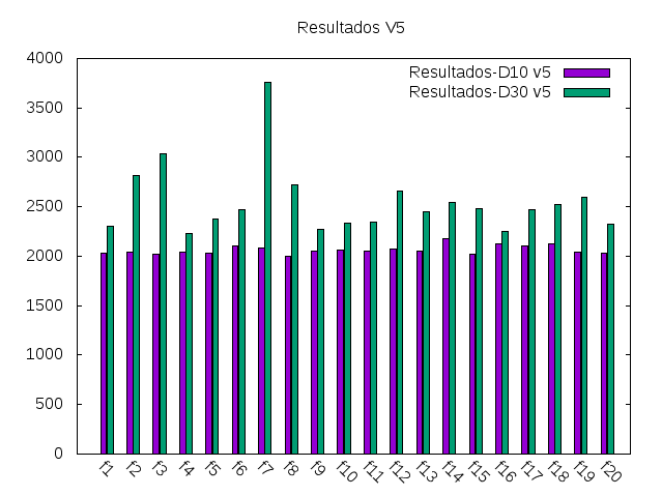


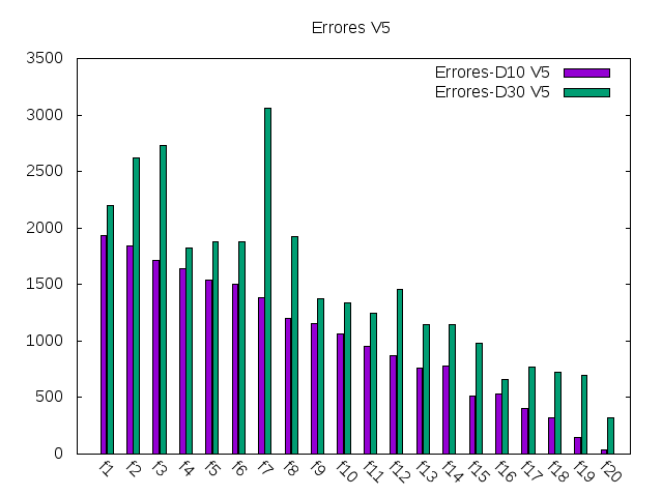




## Características

- Incorporación de Differential Evolution.

# Quinta versión

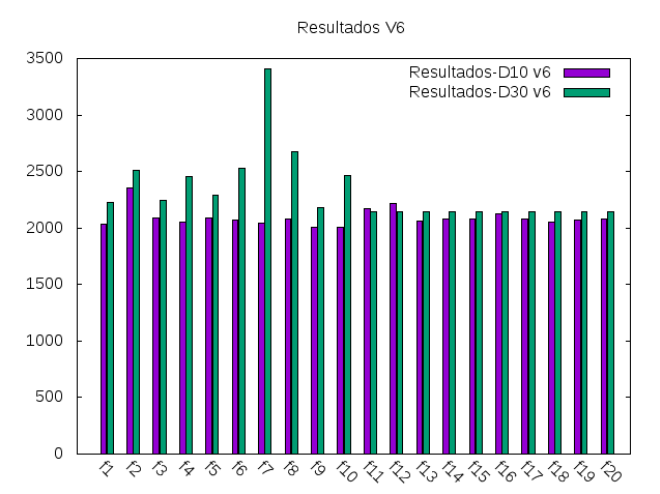


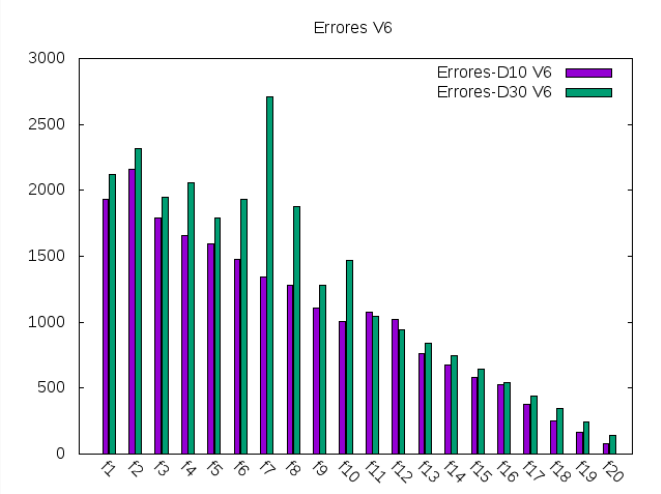




## Características

- Sustitución de Solis Wets por CMAES.

# Sexta versión

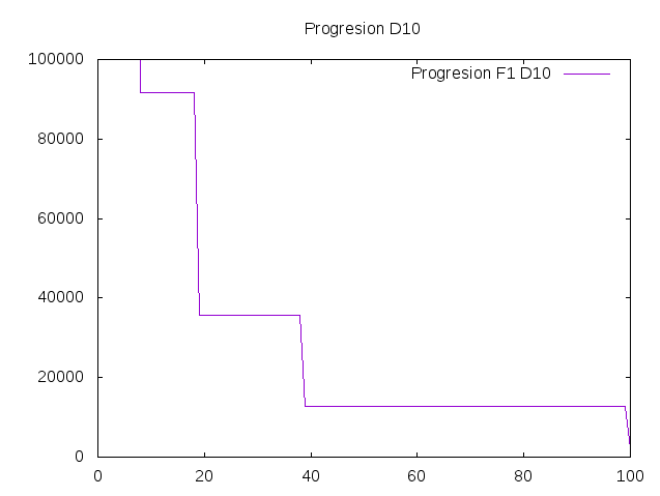


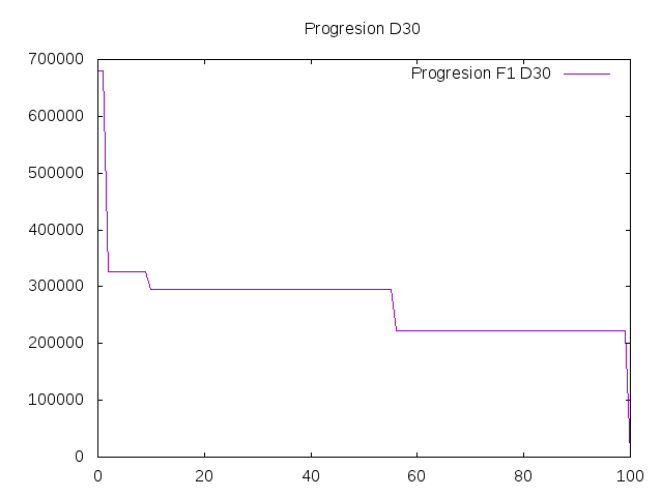




## Características

- Aproximación a un vector aleatorio en vez de a una ballena aleatoria.

# Progresión de las ballenas




## Comentarios

- La mejora se produce cuando DE o CMAES se ejecutan.
- Aprovechamos las iteraciones del algoritmo porque mejoramos en general.
- Este esquema es sólo para la función 1 ya que se parecía en el resto de funciones.

# Comparativa con los algoritmos

| Posición | Clasificación D10 | Error | Clasificación D30 | Error |
|---|---|---|---|---|
| 1 | L-SHADE | 81.2756 | iCMAES-ILS | 1440 |
| 2 | D-SHADE | 112.8402 | L-SHADE | 1470 |
| 3 | CoDE | 114.0602 | D-SHADE | 1570 |
| 4 | SHADE11 | 118.4516 | GaAPADE | 2180 |
| 5 | rmalschcma | 138.2605 | NBIPOP-aCMA-ES | 2410 |
| 6 | MVMO | 140.7138 | MVMO | 2660 |
| 7 | dynNP-jDE | 165.5208 | UMOEAS | 2680 |
| 8 | JADE | 166.6920 | SHADE11 | 3050 |
| 9 | UMOEAS | 176.1225 | rmalschcma | 3180 |
| 10 | iCMAES-ILS | 185.9477 | CMLSP | 4360 |
| 11 | GaAPADE | 250.8654 | RSDE | 6070 |
| 12 | SaDE | 270.1111 | JADE | 15100 |
| 13 | NBIPOP-aCMA-ES | 281.6920 | POBL_ADE | 23800 |
| 14 | EPSDE | 409.9321 | **WOA** | 25400 |
| 15 | RSDE | 445.5327 | CoDE | 29800 |
| 16 | DEBin | 577 | dynNP-jDE | 49500 |
| 17 | DEexp | 632 | EPSDE | 74300 |
| 18 | CMLSP | 737.2535 | OptBees | 118000 |
| 19 | OptBees | 2177.5830 | DEBin | 119000 |
| 20 | FWA-DM | 5712.6597 | FWA-DM | 286000 |
| 21 | POBL_ADE | 19230.3251 | SaDE | 314000 |
| 22 | **WOA** | 20845.9416 | DEexp | 321000 |
| 23 | SOO+BOBYQA | 22242.966 | NRGA | 839000 |
| 24 | NRGA | 56303.5641 | SOO+BOBYQA | 2760000 |
| 25 | SOO | 11963128 | SOO | 244000000 |
| 26 | FCDE | 1719329.8 | b3e3pbest | 534000000 |
| 27 | FERDE | 144238862 | FCDE | 1800000000 |
| 28 | b3e3pbest | 227017644 | FERDE | 3640000000 |

# Comparativa con el resto de algoritmos

**Comentarios**

- En dimensión 10 obtenemos malos resultados.
- En dimensión 30 obtenemos resultados mucho más aceptables superando a DE.
- El esquema de exploración con CMAES y DE añaden valor sobre el DE básico.
- Casi todos los algoritmos que han competido en CEC2014 están por encima.

# Conclusiones

# Conclusiones

## Conclusiones

- Buena idea teórica pero no es eficaz en la práctica.
- La exploración lineal y espiral no funciona bien por sí sola.
- El algoritmo final ha acabado siendo un esquema CMAES-DE con reinicio de la población con una cierta probabilidad.
- No se encuentra en el estado del arte.
- Funciona mejor en dimensión 30 que en dimensión 10.