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.

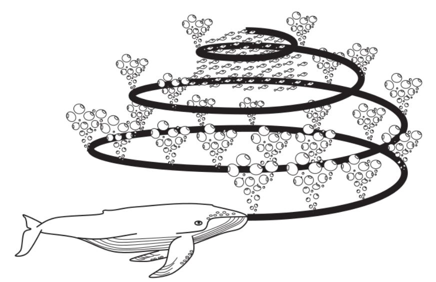
Fases de la caza

Fases de la caza

- Exploración para encontrar presas.
- Caza de presas.

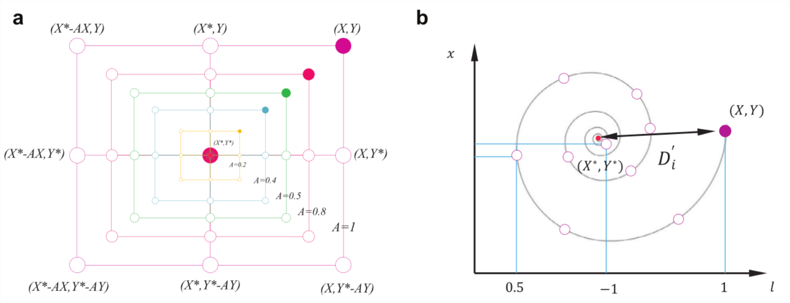
Fases de la caza

- Exploración para encontrar presas.
- Caza de presas.

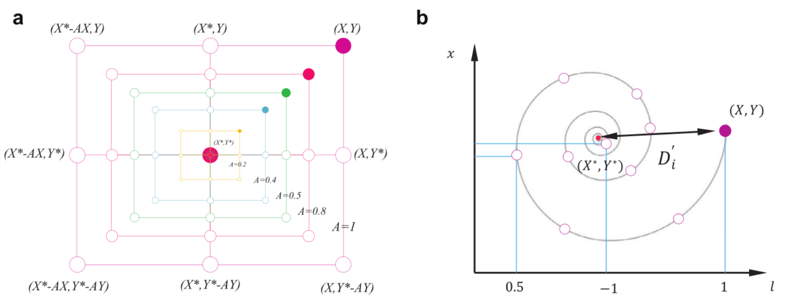


Modelo matemático

Aproximación a la presa

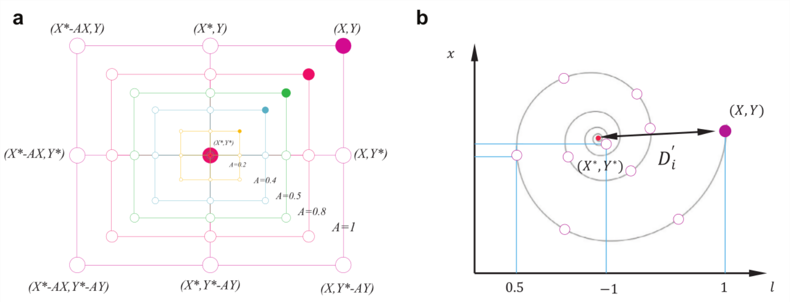


Aproximación a la presa



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

Aproximación a la presa

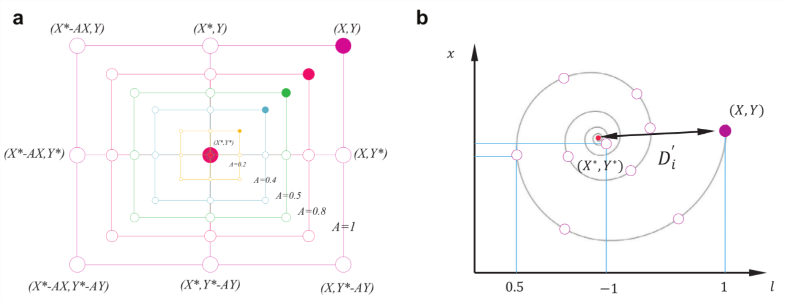


$$D(t) = |\vec{C} \cdot \vec{X^*}(t) - \vec{X}(t)| \qquad \vec{X}(t+1) = \vec{X^*}(t) - \vec{A} \cdot D(t)$$

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

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

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 un vector aleatorio.

Pseudocódigo

```
Initialize the whales population Xi (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 (Xrand)
                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 = ∞
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

Conclusiones

Ideas y preguntas.