Whale Optimization Algorithm (WOA) para el problema de clustering con restricciones

Yábir García Benchakhtir

June 25, 2020

Universidad de Granada

Descripción del problema

Etapas en la caza

Resolvemos el problema del agrupamiento con restricciones

- Partimos de un conjunto de puntos
- Una serie de restricciones entre ellos

Etapas en la caza

Resolvemos el problema del agrupamiento con restricciones

- Partimos de un conjunto de puntos
- Una serie de restricciones entre ellos

Nuestro objetivo es agruparlos en conjuntos de manera que haya relación entre los elementos de un mismo conjunto minimizando el número de restricciones incumplidas.

Algunas definiciones importantes

- Centroide de un cluster
$$\mu_i = \frac{1}{|c_i|}\sum_{x \in c_i} x \quad \text{con } c_i \in \mathcal{C} \text{ para todo } i \in \{1, ...k\}$$

Algunas definiciones importantes

- Centroide de un cluster

$$\mu_i = \frac{1}{|c_i|}\sum_{x \in c_i} x \quad \text{con } c_i \in \mathcal{C} \text{ para todo } i \in \{1, ...k\}$$

- Distancia intracluster

$$\bar{c}_i = \frac{1}{|c_i|}\sum_{x \in c_i} ||x - \mu_i||_2 \quad \text{con } c_i \in \mathcal{C} \text{ para todo } i \in \{1, ...k\}$$

Algunas definiciones importantes

- Centroide de un cluster

$$\mu_i = \frac{1}{|c_i|} \sum_{x \in c_i} x \quad \text{con } c_i \in \mathcal{C} \text{ para todo } i \in \{1, ...k\}$$

- Distancia intracluster

$$\bar{c}_i = \frac{1}{|c_i|} \sum_{x \in c_i} ||x - \mu_i||_2 \quad \text{con } c_i \in \mathcal{C} \text{ para todo } i \in \{1, ...k\}$$

- A partir de esta definimos

$$\bar{C} = \frac{1}{k} \sum_{c_i \in \mathcal{C}} \bar{c}_i$$

Algunas definiciones importantes

- Centroide de un cluster
$$\mu_i = \frac{1}{|c_i|}\sum_{x \in c_i} x \quad \text{con } c_i \in \mathcal{C} \text{ para todo } i \in \{1, ...k\}$$

- Distancia intracluster
$$\bar{c}_i = \frac{1}{|c_i|}\sum_{x \in c_i} ||x - \mu_i||_2 \quad \text{con } c_i \in \mathcal{C} \text{ para todo } i \in \{1, ...k\}$$

- A partir de esta definimos
$$\bar{C} = \frac{1}{k}\sum_{c_i \in \mathcal{C}} \bar{c}_i$$

- Constante que nos va a decir como de importantes son las restricciones
$$\lambda = \frac{\lceil d \rceil}{|R|}$$

Nuestro objetivo será optimizar la función

$$f = \bar{C} + \lambda * \text{infeasibility}$$

Caza de la ballena jorobada

Etapas en la caza

- Movimiento rectilineo en busca de una presa
- Ataque sobre la presa

Movimiento rectilineo

La ballena se desplaza por el medio buscando una presa a la que atacar

$$X_i(t+1) = X^*(t) - A \cdot D_i^1 \tag{1}$$

$$D_i^1 = ||CX^*(t) - X_i(t)|| \tag{2}$$

Movimiento rectilineo

La ballena se desplaza por el medio buscando una presa a la que atacar

$$X_i(t+1) = X^*(t) - A \cdot D_i^1 \tag{1}$$

$$D_i^1 = ||CX^*(t) - X_i(t)|| \tag{2}$$

con

$$A = 2a \cdot r - a$$

$$C = 2r$$

siendo $r \in [0,1]^d$ un vector aleatorio y $a \in [0,2]^d$ constante que se hace decrecer de manera lineal a lo largo de los distintos pasos del algoritmo mediante la ecuación

$$a(t) = 2 - 2\frac{t}{\text{max\_evaluaciones}}$$

Ataque sobre la presa

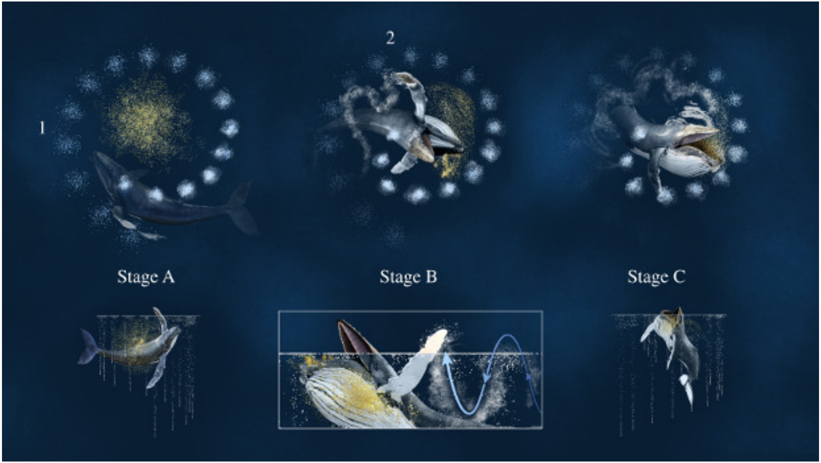

Figure 1: Fases de la caza de la ballena jorobada

1

[1]https://nmas1.org/news/2019/10/17/ballena-aletas-peces-boca

Ataque sobre la presa

Para modelas la fase de ataque sobre la presa usamos

$$\begin{cases} X_i(t+1) & = e^{bl} cos(2\pi l) D_2 + X^* \\ D_i^2 & = ||X^*(t) - X_i(t)|| \end{cases} \tag{3}$$

Resumen de los movimeintos

$$X_i(t+1) = \begin{cases} X^*(t) - A \cdot D_i^1 & p < \frac{1}{2} \\ e^{bl} cos(2\pi l) D_i^2 + X^* & p \geq \frac{1}{2} \end{cases}$$

Pseudo-código de la metaheurística

Algorithm 1 Whale optimization Algorithm

```
procedure WOA(max_evaluaciones)
    whales: Inicializar un conjunto de ballenas con k centroides aleatorios
    Evaluamos las diferentes ballenas usando nuestra métrica
    Seleccionamos X* la mejor ballena
    Guardamos la mejor solución encontrada al problema
    evaluaciones ← 0
    while evaluaciones < max_evaluaciones do
        actualizamos el parametro a
        for agente en la lista de ballenas do
            Determinar p y calcular A y C.
            if p < 0,5 then
                if |A| < 1 then
                    Movemos la ballena usando el movimiento rectilineo (1)
                else if |A| > 1 then
                    Movimiento rectilineo usando una ballena aleatorio (1)
                end if
            else
                Movimiento en espiral utilizando (2)
            end if
        end for
        Comprobar si alguna solución se ha salido de los limites del problema
        Incrementar evaluaciones en el número de evaluaciones correspondiente
        Evaluar las soluciones encontradas y actualizar la mejor ballena
        if la mejor ballena es la mejor solución encontrada then
            Actualizar la mejor solución encontrada hasta el momento
        end if
    end while
    return Construir la solución asociada a la mejor ballena encontrada
end procedure
```