

Reptile Search Algorithm

Dr. Broderick Crawford Labrín

Pontificia Universidad Católica de Valparaíso

Reptile Search Algorithm (RSA)

- Fue desarrollada por Krishna Gopal Dhal et. al. en el año 2022 ^a.
- Es una metaheurística basado en población diseñada para resolver problemas de optimización continuos.
- Sus soluciones (individuos) iniciales se generan aleatoriamente y se van alterando bajo un conjunto de reglas de movimiento con criterios estocásticos.

^a *Reptile Search Algorithm: Theory, Variants, Applications, and Performance Evaluation*, Springer Link (2023)

- Ecuaciones de movimientos general

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,D} \\ x_{2,1} & x_{2,2} & \dots & x_{2,D} \\ \dots & \dots & \dots & \dots \\ x_{N,1} & x_{N,2} & \dots & x_{N,D} \end{bmatrix} \quad (1)$$

$$x_{i,j} = rand \cdot (UB - LB) + LB \quad (2)$$

- Donde:
 - X representa un colección de soluciones individuales
 - N representa el tamaño de la población
 - D representa la dimensión de la población
 - $x_{i,j}$ representa la j^{va} posición del i^{va} solución
 - $k = 1, 2, \dots, n$

- Ecuaciones de movimiento (Exploración)

$$M_{(x_i)} = \frac{1}{D} \sum_{j=1}^D x_{i,j} \quad (3)$$

$$P_{i,j} = \alpha + \frac{x_{i,j} - M_{x_i}}{UB_j - LB_j + \epsilon} \quad (4)$$

- Donde:

- $M_{(x_i)}$ es el promedio de las soluciones
- P representa la diferencia en porcentaje entra la mejor solucion y la actual
- $Best_j$ representa la mejor posición del individuo más optimo

- Ecuaciones de movimiento (Exploración)

$$ES(t) = 2 \cdot r_3 \cdot \left(1 - \frac{1}{MaxIt}\right) \quad (5)$$

$$R_{i,j} = \frac{Best_j(t) - x_{r2,j}}{Best_j(t) + \varepsilon} \quad (6)$$

- Donde:
 - $ES(t)$ representa el ratio de probabilidad con un rango de -2 a 2
 - r_3 es un número random [-1,1]
 - $R_{i,j}$ se utiliza para minimizar la región de búsqueda con:
 - r_2 siendo un número random [1,N]
 - ε es un pequeño número

- Ecuaciones de movimiento (Exploración)

$$\eta_{i,j} = Best_j(t) \cdot P_{i,j}(t) \quad (7)$$

$$X_{i,j}^{(t+1)} \left\{ \begin{array}{l} Best_j(t) - \eta_t \cdot \beta - R_{i,j}(t) \cdot rand, t \leq \frac{MaxIt}{4} \\ Best_j(t) \cdot x_{r1,j}(t) \cdot ES(t) \cdot rand, t \leq 2\frac{MaxIt}{4} \text{ y } t > \frac{MaxIt}{4} \end{array} \right\} \quad (8)$$

- Donde:

- $\eta_{i,j}$ representa el operador de caza de la k -ésima posición en la i -ésima solución
- X representa un colección de soluciones individuales
- $x_{i,j}$ representa la j^{va} posición del i^{va} solución
 - $k = 1, 2, \dots, n$

Algorithm 1 Reptile Search Algorithm

Input: Population $X = \{x_1, x_2, \dots, x_N\}$

Output: Updated population $X' = \{x'_1, x'_2, \dots, x'_n\}$ and Best

```
1: Initialize random population X, constant  $\alpha$ , constant  $\beta$  and constant  $\epsilon$ 
2: for it = 1 to MaxIt do
3:     calculate the fitness of each X
4:     select Best
5:     r3 = random(-1,1)
6:     ES = 2*r3*(1-(1/maxIter))
7:     for i=0 to N do
8:         for j=0 to Dim do
9:             r2 = random(0,N-1)
10:            R = (Best - Xr2,j)/(Bestj+ $\epsilon$ )
11:            P =  $\alpha + (X_{i,j} - \bar{X}_i) / (UB_j - LB_j + \epsilon)$ 
12:             $\eta = Best_j * P$ 
13:            rand = random()
14:            if it < MaxIter/4
15:                Xi,j = Bestj -  $\eta * \beta - R * rand$ 
16:            else if it < (2*MaxIter)/4 and it  $\geq$  MaxIter/4
17:                r1 = random(0,N-1)
18:                Xi,j = Bestj * Xr1,j * ES * rand
19:            else if it < (MaxIter*3)/4 and it  $\geq$  (2*it)/4
20:                Xi,j = Bestj * P * rand
21:            Else
22:                Xi,j = Bestj -  $\eta * \epsilon - R * rand$ 
23:
24: return updated population X' and Best
```

Considerando

$$\text{Min } z = x_1^2 + x_2^2 + x_3^2$$

Sujeto a

$$x_1, x_2, x_3 \in [-100, 100]$$

Configuración inicial de RSA:

- Tamaño de la población: 2 individuos
- Número máximo de iteraciones: 100 iteraciones
- $\alpha = 0.1$
- $\beta = 0.1$
- $\varepsilon = 1e^{-10}$

RSA: Ejemplo práctico - soluciones iniciales

Soluciones iniciales:

ind 1: [-35.1045, 4.535, -36.8781] / fitness: 2612.8903

ind 2: [71.0945, 21.6292, -94.0776] / fitness: 14372.8538

Mejor solución:

ind 1: [-35.1045, 4.535, -36.8781] / fitness: 2612.8903

RSA: Ejemplo práctico - iter 1

Ecuaciones generales de la iteración 1:

$$r_3 = \text{random}(-1, 1) = 1$$

$$ES(t) = 2 \cdot r_3 \cdot \left(1 - \frac{1}{MaxIt}\right) = 2 \cdot 1 \cdot \left(1 - \frac{1}{100}\right) = 1,9800$$

Ecuación general RSA:

$$r_2 = \text{random}(0, N - 1)$$

$$R_{i,j} = \frac{Best_j(t) - x_{r_2,j}}{Best_j(t) + \varepsilon}$$

$$P_{i,j} = \alpha + \frac{x_{i,j} - M_{xi}}{UB_j - LB_j + \varepsilon}$$

$$\eta_{i,j} = Best_j(t) \cdot P_{i,j}(t)$$

$$iter < MaxIt$$

$$X_{i,j}^{(t+1)} = Best_j(t) - \eta_t \cdot \beta - R_{i,j}(t) \cdot rand$$

RSA: Ejemplo práctico - iter 1 - ind 1 - dim 1

$$r_2 = \text{random}(0, 2 - 1) = 0$$

$$R_{1,1} = \frac{-35,1045 + 35,1045}{-35,1045 + 0,0000} = 0,0000$$

$$P_{1,1} = 0,1 + \frac{-35,1045 + 22,4825}{100 + 100 + 0,0000} = 0,0369$$

$$\eta_{1,1} = -35,1045 \cdot 0,0369 = -1,2950$$

$$\text{rand} = \text{random}() = 0,9515$$

$$it < \frac{\text{maxIter}}{4}$$

$$1 < 25,0000$$

$$X_{1,1}^2 = -35,1045 + 1,2950 \cdot 0,1 + 0,0000 \cdot 0,9515 = -34,9750$$

RSA: Ejemplo práctico - iter 1 - ind 1 - dim 2

$$r_2 = \text{random}(0, 2 - 1) = 0$$

$$R_{1,2} = \frac{4,5350 - 4,5350}{4,5350 + 0,0000} = 0,0000$$

$$P_{1,2} = 0,1 + \frac{4,5350 + 22,4394}{100 + 100 + 0,0000} = 0,2349$$

$$\eta_{1,2} = 4,5350 \cdot 0,2349 = 1,0652$$

$$\text{rand} = \text{random}() = 0,4883$$

$$it < \frac{\text{maxIter}}{4}$$

$$1 < 25,0000$$

$$X_{1,2}^2 = 4,5350 - 1,0652 \cdot 0,1 + 0,0000 \cdot 0,4883 = 4,4285$$

$$r_2 = \text{random}(0, 2 - 1) = 1$$

$$R_{1,3} = \frac{-36,8781 + 94,0776}{-36,8781 + 0,0000} = -1,5510$$

$$P_{1,3} = 0,1 + \frac{-36,8781 + 22,4749}{100 + 100 + 0,0000} = 0,0280$$

$$\eta_{1,3} = -36,8781 \cdot 0,0280 = -1,0320$$

$$\text{rand} = \text{random}() = 0,6663$$

$$it < \frac{\text{maxIter}}{4}$$

$$1 < 25,0000$$

$$X_{1,3}^2 = -36,8781 + 1,0320 \cdot 0,1 + 1,5510 \cdot 0,6663 = -35,7414$$

$$r_2 = \text{random}(0, 2 - 1) = 1$$

$$R_{2,1} = \frac{-35,1045 - 71,0945}{-35,1045 + 0,0000} = 3,0252$$

$$P_{2,1} = 0,1 + \frac{71,0945 + 0,4513}{100 + 100 + 0,0000} = 0,4577$$

$$\eta_{2,1} = -35,1045 \cdot 0,4577 = -16,0684$$

$$\text{rand} = \text{random}() = 0,2777$$

$$it < \frac{\text{maxIter}}{4}$$

$$1 < 25,0000$$

$$X_{2,1}^2 = -35,1045 + 16,0684 \cdot 0,1 - 3,0252 \cdot 0,2777 = -34,3376$$

$$r_2 = \text{random}(0, 2 - 1) = 0$$

$$R_{2,2} = \frac{4,5350 - 4,4285}{4,5350 + 0,0000} = 0,0235$$

$$P_{2,2} = 0,1 + \frac{21,6292 + 35,5954}{100 + 100 + 0,0000} = 0,3861$$

$$\eta_{2,2} = 4,5350 \cdot 0,3861 = 1,7511$$

$$\text{rand} = \text{random}() = 0,9399$$

$$it < \frac{\text{maxIter}}{4}$$

$$1 < 25,0000$$

$$X_{2,2}^2 = 4,5350 - 1,7511 \cdot 0,1 - 0,0235 \cdot 0,9399 = 4,3378$$

$$r_2 = \text{random}(0, 2 - 1) = 1$$

$$R_{2,3} = \frac{-36,8781 + 94,0776}{-36,8781 + 0,0000} = -1,5510$$

$$P_{2,3} = 0,1 + \frac{-36,8781 + 22,4749}{100 + 100 + 0,0000} = -0,1636$$

$$\eta_{2,3} = -36,8781 \cdot 0,0280 = 6,0330$$

$$\text{rand} = \text{random}() = 0,1274$$

$$it < \frac{\text{maxIter}}{4}$$

$$1 < 25,0000$$

$$X_{2,3}^2 = -36,8781 + 1,0320 \cdot 0,1 + 1,5510 \cdot 0,6663 = -37,2838$$

RSA: Ejemplo práctico - validación restricciones

Restricción: $x_1, x_2, x_3 \in [-100, 100]$

Soluciones obtenidas en la iteración 1:

ind 1: $[-34.975, 4.4285, -35.7414]$, infactibles: 0

ind 2: $[-34.3377, 4.3379, -37.2838]$, infactibles: 0

Reparación de soluciones:

ind 1: $[-34.975, 4.4285, -35.7414]$ / fitness: 2520.3145

ind 2: $[-34.3377, 4.3379, -37.2838]$ / fitness: 2587.9765

Mejor solución:

ind 1: $[-34.975, 4.4285, -35.7414]$ / fitness: 2520.3145

RSA: Ejemplo práctico - iter 2

Ecuaciones generales de la iteración 2:

$$r_3 = \text{random}(-1, 1) = -1$$

$$ES(t) = 2 \cdot r_3 \cdot \left(1 - \frac{1}{\text{MaxIt}}\right) = 2 \cdot -1 \cdot \left(1 - \frac{1}{100}\right) = -1,9800$$

Ecuación general RSA:

$$r_2 = \text{random}(0, N - 1)$$

$$R_{i,j} = \frac{\text{Best}_j(t) - x_{r_2,j}}{\text{Best}_j(t) + \varepsilon}$$

$$P_{i,j} = \alpha + \frac{x_{i,j} - M_{xi}}{UB_j - LB_j + \varepsilon}$$

$$\eta_{i,j} = \text{Best}_j(t) \cdot P_{i,j}(t)$$

$$\text{iter} < \text{MaxIt}$$

$$X_{i,j}^{(t+1)} = \text{Best}_j(t) - \eta_t \cdot \beta - R_{i,j}(t) \cdot \text{rand}$$

RSA: Ejemplo práctico - iter 2 - ind 1 - dim 1

$$r_2 = \text{random}(0, 2 - 1) = 1$$

$$R_{1,1} = \frac{-34,9750 + 34,3377}{-34,9750 + 0,0000} = 0,0182$$

$$P_{1,1} = 0,1 + \frac{-34,9750 + 22,0960}{100 + 100 + 0,0000} = 0,0356$$

$$\eta_{1,1} = -36,8781 \cdot 0,0280 = -1,2453$$

$$\text{rand} = \text{random}() = 0,9282$$

$$it < \frac{\text{maxIter}}{4}$$

$$2 < 25,0000$$

$$X_{1,1}^3 = -34,9750 + 1,2453 \cdot 0,1 - 0,0182 \cdot 0,9282 = -34,8674$$

RSA: Ejemplo práctico - iter 2 - ind 1 - dim 2

$$r_2 = \text{random}(0, 2 - 1) = 1$$

$$R_{1,2} = \frac{4,4285 - 4,3379}{4,4285 + 0,0000} = 0,0205$$

$$P_{1,2} = 0,1 + \frac{4,4285 + 22,0601}{100 + 100 + 0,0000} = 0,2324$$

$$\eta_{1,2} = 4,4285 \cdot 0,2324 = 1,0294$$

$$\text{rand} = \text{random}() = 0,1672$$

$$it < \frac{\text{maxIter}}{4}$$

$$2 < 25,0000$$

$$X_{1,2}^3 = 4,4285 - 1,0294 \cdot 0,1 - 0,0205 \cdot 0,1672 = 4,3221$$

$$r_2 = \text{random}(0, 2 - 1) = 1$$

$$R_{1,3} = \frac{-35,7414 + 37,2838}{-35,7414 + 0,0000} = -0,0432$$

$$P_{1,3} = 0,1 + \frac{-35,7414 + 22,0956}{100 + 100 + 0,0000} = 0,0318$$

$$\eta_{1,3} = -35,7414 \cdot 0,0318 = -1,1355$$

$$\text{rand} = \text{random}() = 0,4192$$

$$it < \frac{\text{maxIter}}{4}$$

$$2 < 25,0000$$

$$X_{1,3}^3 = -35,7414 + 1,1355 \cdot 0,1 + 0,0432 \cdot 0,4192 = -35,6097$$

RSA: Ejemplo práctico - iter 2 - ind 2 - dim 1

$$r_2 = \text{random}(0, 2 - 1) = 0$$

$$R_{2,1} = \frac{-34,9750 + 34,8674}{-34,9750 + 0,0000} = 0,0031$$

$$P_{2,1} = 0,1 + \frac{-34,3377 + 22,4279}{100 + 100 + 0,0000} = 0,0405$$

$$\eta_{2,1} = -34,9750 \cdot 0,0405 = -1,4148$$

$$\text{rand} = \text{random}() = 0,7818$$

$$it < \frac{\text{maxIter}}{4}$$

$$2 < 25,0000$$

$$X_{2,1}^3 = -34,9750 + 1,4148 \cdot 0,1 - 0,0031 \cdot 0,7818 = -34,8359$$

$$r_2 = \text{random}(0, 2 - 1) = 1$$

$$R_{2,2} = \frac{4,4285 - 4,3379}{4,4285 + 0,0000} = 0,0205$$

$$P_{2,2} = 0,1 + \frac{4,3379 + 22,5940}{100 + 100 + 0,0000} = 0,2347$$

$$\eta_{2,2} = 4,4285 \cdot 0,2347 = 1,0392$$

$$\text{rand} = \text{random}() = 0,2563$$

$$it < \frac{\text{maxIter}}{4}$$

$$2 < 25,0000$$

$$X_{2,2}^3 = 4,4285 - 1,0392 \cdot 0,1 - 0,0205 \cdot 0,2563 = 4,3193$$

$$r_2 = \text{random}(0, 2 - 1) = 0$$

$$R_{2,3} = \frac{-35,7414 + 35,6098}{-35,7414 + 0,0000} = 0,0037$$

$$P_{2,3} = 0,1 + \frac{-37,2838 + 22,6001}{100 + 100 + 0,0000} = 0,0266$$

$$\eta_{2,3} = -35,7414 \cdot 0,0266 = -0,9501$$

$$\text{rand} = \text{random}() = 0,5691$$

$$it < \frac{\text{maxIter}}{4}$$

$$2 < 25,0000$$

$$X_{2,3}^3 = -35,7414 + 0,9501 \cdot 0,1 - 0,0037 \cdot 0,5691 = -35,6485$$

RSA: Ejemplo práctico - validación restricciones

Restricción: $x_1, x_2, x_3 \in [-100, 100]$

Soluciones obtenidas en la iteración 1:

ind 1: [-34.8674, 4.3222, -35.6098], infactibles: 0

ind 2: [-34.836, 4.3194, -35.6485], infactibles: 0

Reparación de soluciones:

ind 1: [-34.8674, 4.3222, -35.6098] / fitness: 2502.4749

ind 2: [-34.836, 4.3194, -35.6485] / fitness: 2503.0179

Mejor solución:

ind 1: [-34.8674, 4.3222, -35.6098] / fitness: 2502.4749

Ecuaciones generales de la iteración 100:

$$r_3 = \text{random}(-1, 1) = 0$$

$$ES(t) = 2 \cdot r_3 \cdot \left(1 - \frac{1}{MaxIt}\right) = 2 \cdot 0 \cdot \left(1 - \frac{1}{100}\right) = 0,0000$$

Ecuación general RSA:

$$r_2 = \text{random}(0, N - 1)$$

$$R_{i,j} = \frac{Best_j(t) - x_{r_2,j}}{Best_j(t) + \varepsilon}$$

$$P_{i,j} = \alpha + \frac{x_{i,j} - M_{xi}}{UB_j - LB_j + \varepsilon}$$

$$\eta_{i,j} = Best_j(t) \cdot P_{i,j}(t)$$

$$iter < MaxIt$$

$$X_{i,j}^{(t+1)} = Best_j(t) - \eta_t \cdot \beta - R_{i,j}(t) \cdot rand$$

$$r_2 = \text{random}(0, 2 - 1) = 1$$

$$R_{1,1} = \frac{0,0000 - 0,0000}{0,0000 + 0,0000} = 0,0000$$

$$P_{1,1} = 0,1 + \frac{0,0000 - 0,0000}{100 + 100 + 0,0000} = 0,1000$$

$$\eta_{1,1} = 0,0000 \cdot 0,1000 = 0,0000$$

$$\text{rand} = \text{random}() = 0,3323$$

$$X_{1,1}^{101} = 0,0000 - 0,0000 \cdot 0,0000 - 0,0000 \cdot 0,3323 = 0,0000$$

RSA: Ejemplo práctico - iter 100 - ind 1 - dim 2

$$r_2 = \text{random}(0, 2 - 1) = 1$$

$$R_{1,2} = \frac{0,0000 - 0,0000}{0,0000 + 0,0000} = 0,0000$$

$$P_{1,2} = 0,1 + \frac{0,0000 - 0,0000}{100 + 100 + 0,0000} = 0,1000$$

$$\eta_{1,2} = 0,0000 \cdot 0,1000 = 0,0000$$

$$\text{rand} = \text{random}() = 0,4467$$

$$X_{1,2}^{101} = 0,0000 - 0,0000 \cdot 0,0000 - 0,0000 \cdot 0,4467 = 0,0000$$

$$r_2 = \text{random}(0, 2 - 1) = 1$$

$$R_{1,3} = \frac{0,0000 - 0,0000}{0,0000 + 0,0000} = 0,0000$$

$$P_{1,3} = 0,1 + \frac{0,0000 - 0,0000}{100 + 100 + 0,0000} = 0,1000$$

$$\eta_{1,3} = 0,0000 \cdot 0,1000 = 0,0000$$

$$\text{rand} = \text{random}() = 0,4854$$

$$X_{1,3}^{101} = 0,0000 - 0,0000 \cdot 0,0000 - 0,0000 \cdot 0,4854 = 0,0000$$

RSA: Ejemplo práctico - iter 100 - ind 2 - dim 1

$$r_2 = \text{random}(0, 2 - 1) = 0$$

$$R_{2,1} = \frac{0,0000 - 0,0000}{0,0000 + 0,0000} = 0,0000$$

$$P_{2,1} = 0,1 + \frac{0,0000 - 0,0000}{100 + 100 + 0,0000} = 0,1000$$

$$\eta_{2,1} = 0,0000 \cdot 0,1000 = 0,0000$$

$$\text{rand} = \text{random}() = 0,6596$$

$$X_{2,1}^{101} = 0,0000 - 0,0000 \cdot 0,0000 - 0,0000 \cdot 0,6596 = 0,0000$$

RSA: Ejemplo práctico - iter 100 - ind 2 - dim 2

$$r_2 = \text{random}(0, 2 - 1) = 0$$

$$R_{2,2} = \frac{0,0000 - 0,0000}{0,0000 + 0,0000} = 0,0000$$

$$P_{2,2} = 0,1 + \frac{0,0000 - 0,0000}{100 + 100 + 0,0000} = 0,1000$$

$$\eta_{2,2} = 0,0000 \cdot 0,1000 = 0,0000$$

$$\text{rand} = \text{random}() = 0,4807$$

$$X_{2,2}^{101} = 0,0000 - 0,0000 \cdot 0,0000 - 0,0000 \cdot 0,4807 = 0,0000$$

RSA: Ejemplo práctico - iter 100 - ind 2 - dim 3

$$r_2 = \text{random}(0, 2 - 1) = 1$$

$$R_{2,3} = \frac{0,0000 - 0,0000}{0,0000 + 0,0000} = 0,0000$$

$$P_{2,3} = 0,1 + \frac{0,0000 - 0,0000}{100 + 100 + 0,0000} = 0,1000$$

$$\eta_{2,3} = 0,0000 \cdot 0,1000 = 0,0000$$

$$\text{rand} = \text{random}() = 0,4241$$

$$X_{2,3}^{101} = 0,0000 - 0,0000 \cdot 0,0000 - 0,0000 \cdot 0,4241 = 0,0000$$

RSA: Ejemplo práctico - validación restricciones

Restricción: $x_1, x_2, x_3 \in [-100, 100]$

Soluciones obtenidas en la iteración 100:

ind 1: [0, 0, 0], infactibles: 0

ind 2: [0, 0, 0], infactibles: 0

Reparación de soluciones:

ind 1: [0, 0, 0] / fitness: 0.0000

ind 2: [0, 0, 0] / fitness: 0.0000

Mejor solución:

ind 1: [0. 0. 0.] / fitness: 0.0000