



Instituto Politécnico Nacional
Escuela Superior de Cómputo
Centro de Investigación en
Computación



Ingeniería en Inteligencia Artificial, Mexican Axolotl Optimization
Fecha: 4 de junio de 2025

MEXICAN AXOLOTL OPTIMIZATION

Presenta

Angeles López Erick Jesse¹

Disponible en:

github.com/JesseAngeles/Metaheuristicas

Resumen

RESUMEN

Palabras clave: PALABRAS CLAVE

¹eangeles1700@alumno.ipn.mx

Índice general

Introducción	3
1. Mexican Axolotl Optimization	4
1.1. Modelo natural	4
1.2. Modelo artificial	4
1.2.1. Transition	5
1.2.2. Injury and restoration	5
1.2.3. Reproduction	6
1.2.4. Assortment	7
2. Problemas CEC 2017	8
3. Código	9
4. Resultados	10
Conclusiones	11
Referencias	12
Anexos	13

Introducción

Capítulo 1

Mexican Axolotl Optimization

Los dioses mexicas se reunieron en Teotihuacan para crear el universo ofreciendo su propia vida en sacrificio. Dioses como Huitzilopochtli, Xochipilli y Tezcatlipoca tomaron el sacrificio, sin embargo, Xolotl, victima del miedo, huyo del ritual.

Xolotl se transformo en múltiples criaturas para evitar se atrapado por el viento, se transformo en diferentes criaturas pero seguía siendo encontrado, por lo que opto por arrojarse al lago convirtiéndose en un anfibio con branquias en forma de cuernos, un axolote. Pudo sobrevivir algunos días en el lago pero finalmente fue atrapado por el viento y llevado al ritual para dar movimiento al universo [1].

1.1. Modelo natural

El axolote mexicano o *Ambystoma mexicanum* es una especie endémica del lago de Xochimilco, son salamandras que nunca superan su fase larvaria. Tiene branquias que brotan de su cabeza, patas palmeadas, una aleta dorsal, cola, pulmones funcionales, camuflaje y una sonrisa.

Los axolotes son un tema de investigación por biólogos dado a su capacidad de regenerar extremidades y órganos sin cicatrices permanentes [2].

1.2. Modelo artificial

El algoritmo *Mexican Axolotl Optimization* (**MAO**) busca imitar el comportamiento de estos anfibios para encontrar la mejor solución en un espacio de búsqueda dadas las siguientes analogías:

- Los axolotes son soluciones de problemas y sus órganos y extremidades las dimensiones de la solución.

- El lago en donde viven es el espacio de búsqueda.
- El camuflaje del axolote, cuya efectividad depende del estado de búsqueda, determina la capacidad de sobrevivir, es decir, su aptitud.

En otras palabras, sea O un problema de optimización numerica cuyas elementos sea vectores de dimensión D acotados en el rango $[\text{mín}_i, \text{máx}_i]$, los axolotes representan un conjunto de soluciones de tamaño np : $P = \{S_1, \dots, S_{np}\}$ donde $O(S_j) \in \mathbb{R} \forall j \in \{1, \dots, np\}$.

El algoritmo MAO es un proceso iterativo de 4 etapas definida por el acronimo *TIRA*: transición de larva a adulto (*Transition*), lesión y restauración (*Injury and restoration*), reproducción (*Reproduction*) y variedad (*Assortment*).

1.2.1. Transition

La población de axolotes se inicia de manera aleatoria y a cada individuo se le asigna un genero, obteniendo dos subconjuntos. Los individuos de cada grupo transicionan en el agua de larvas a adultos ajustando el color de las partes de su cuerpo para parecerse mas al mejor de cada grupo, como se muestra en el algoritmo 1

Algorithm 1 Transition

Input constante de diferenciación α , Población P , Función de optimización O

Output Población actualizada P'

- 1: El mejor axolote: $p_{best} = \text{Best}(P)$
 - 2: **for** $p_j \in P$ **do**
 - 3: Probabilidad inversa de transición: $pp_j = \frac{O(p_j)}{\sum O(p_i)}$
 - 4: **if** $pp_j < r$ **then**
 - 5: Se aproxima al mejor: $p_{j,i} = p_{j,i} + (p_{best,i} - p_{j,i}) \cdot \alpha$
 - 6: **else**
 - 7: Se mueve de forma aleatoria: $p_{j,i} = \text{mín}_i + (\text{máx}_i - \text{mín}_i) * r_i$
 - 8: **end if**
 - 9: **end for**
-

La habilidad de los axolotes de cambiar de color esta limitada, por lo que unicamente se aproximan a la nueva solución, ya que no se busca que se adapten de forma perfecta al mejor de ellos. Cuando los individuos son diferentes mucho de la mejor solución, medida que se obtiene con la probabilidad inversa, se mueven de manera aleatoria para explorar el espacio.

1.2.2. Injury and restoration

Mientras los axolotes se mueven en el agua pueden sufrir accidentes y ser lastimados. En esta fase los axolotes pierden partes que posteriormente regene-

ran de forma aleatoria, para esto, se consideran ambas probabilidades donde la primera actúa sobre los axolotes que serán lastimados y la segunda sobre las partes del cuerpo que serán regeneradas, esto se aplica para ambos grupos como se muestra en el algoritmo 2.

Algorithm 2 Injury and Restoration

Input Población P , probabilidad de daño dp , probabilidad de regeneración rp

Output Población actualizada P'

```

1: for  $p_j \in P$  do
2:   if  $r \leq dp$  then
3:     for  $i \in D$  do
4:       if  $r \leq rp$  then
5:          $p_{j,i} = \text{mín}_i + (\text{máx}_i - \text{mín}_i) * r_i$ 
6:       end if
7:     end for
8:   end if
9: end for

```

1.2.3. Reproduction and Assortment

Los machos y las hembras se reproducen dejando un par de huevos que contienen una mezcla del material genético de los padres de forma uniforme. Se genera una cadena aleatoria de tamaño D y si es un 0 el primer hijo hereda el gen del padre y el dos de la madre, si es un 1 el primer hijo hereda del gen de la madre y el dos del padre, como se muestra en la figura 1.1.

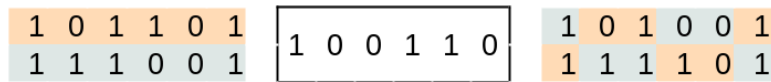


Figura 1.1: Cruza uniforme

El algoritmo 3 selecciona, entre un rango de machos, con quien reproducirse mediante un torneo de tamaño k con cada hembra, cuyos huevos heredan el material genético. Finalmente, entre los 4 individuos se seleccionan los dos mejores y se les asigna el genero de hembra y macho al primero y segundo mejor respectivamente.

Algorithm 3 Reproduction

Input Población P , probabilidad de daño dp , probabilidad de regeneración rp **Output** Población actualizada P'

```

1: for  $f \in F$  do
2:    $m = \{Best(MC) | MC \subset M \wedge |MC| = k\}$ 
3:    $egg_1$ 
4:    $egg_2$ 
5:   for  $i \in D$  do
6:     if  $r \leq 0,5$  then
7:        $huevo_{1,i} = m_i$ 
8:        $huevo_{2,i} = f_i$ 
9:     else
10:       $huevo_{1,i} = f_i$ 
11:       $huevo_{2,i} = m_i$ 
12:    end if
13:  end for
14: end for
15:  $V = Sort(function = O, \{f, m, huevo_1, huevo_2\})$ 
16:  $f = V[0]$ 
17:  $m = V[1]$ 

```

Capítulo 2

Problemas CEC 2017

Capítulo 3

Código

Capítulo 4

Resultados

Conclusiones

Bibliografía

[1]

[2]

Anexos