



Colegio de Ciencias e Ingeniería

**Inteligencia Artificial
NRC: 4004**

Particle Swarm Optimization

Integrantes:

Gabriel Oña– 320597

26 de septiembre del 2023

Algoritmos Particle Swarm Optimization

En este trabajo se buscó minimizar una función algebraica de dos variables que se caracteriza por ser difícil la obtención de un valor mínimo por métodos empíricos. Por lo tanto otro algoritmo sumamente interesante que ayuda a encontrar el valor $[x,y]$ el cual minimiza la función es Particle Swarm Optimization.

La propuesta de resolución es caracterizar en generar un arreglo S de partículas que se van a distribuir alrededor de la superficie de forma aleatoria. Una vez realizado esto, se inicializa el vector de velocidades que van a tomar las N partículas en el plano de forma aleatoria o en 0. Luego se determina el arreglo P con las mejores posiciones de cada partícula. Al iniciar este arreglo P es igual a S ya que para cada partícula su valor inicial es el mejor intento de minimización que existe. Adicionalmente, se busca el índice para el cual $P[g]$ es el mejor valor global que se aproxima a la minimización.

Se itera dependiendo de las épocas deseadas y se va actualizando el vector P con los mejores estados individuales. El índice g en la posición del mejor valor global de P. El vector de velocidades individuales que presenta dos componentes importantes. La primera es un peso y una variable aleatoria uniforme en el rango 0, 1 que indica cuanta importancia se le va a dar al factor cognitivo del sistema. Mientras que el segundo componente, indica el peso hacia el aspecto social al evaluar el mejor de la sociedad.

Se obtuvieron los siguientes resultados:

Optimizar:

$$f(x, y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$$

En el intervalo: $-4.5 \leq x, y \leq 4.5$

Figura 1. Función para minimizar

Sin velocidad inicial, se puede apreciar un comportamiento continuo y evolutivo durante el paso del tiempo.

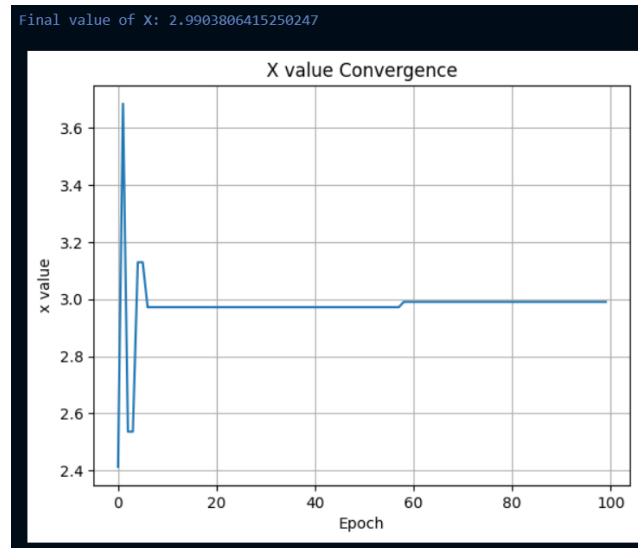


Figura 2. Convergencia x

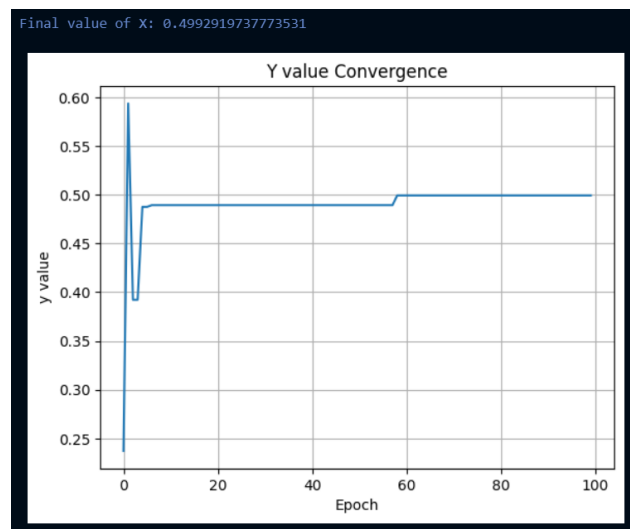


Figura 2. Convergencia y

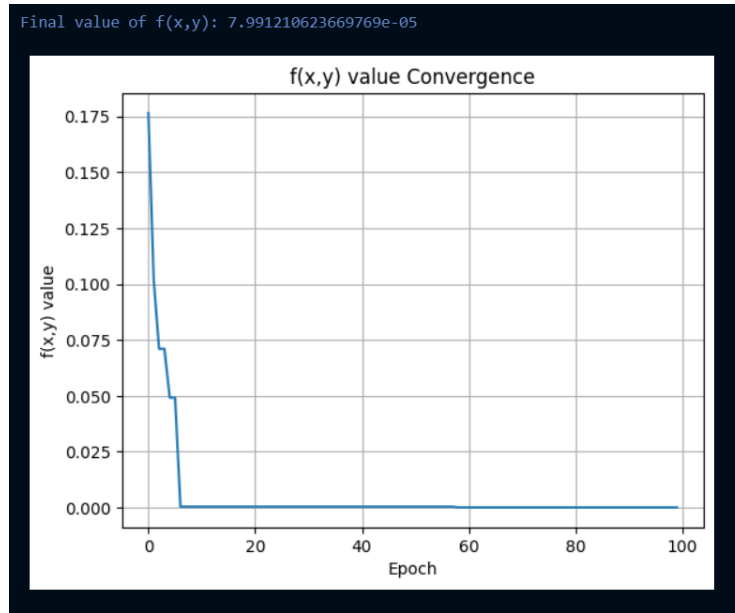


Figura 3. Convergencia $f(x,y)$

Mientras que al implementar una solución que da velocidad inicial distinta de 0 bajo una probabilidad uniforme, la respuesta del sistema es mucho más brusca.

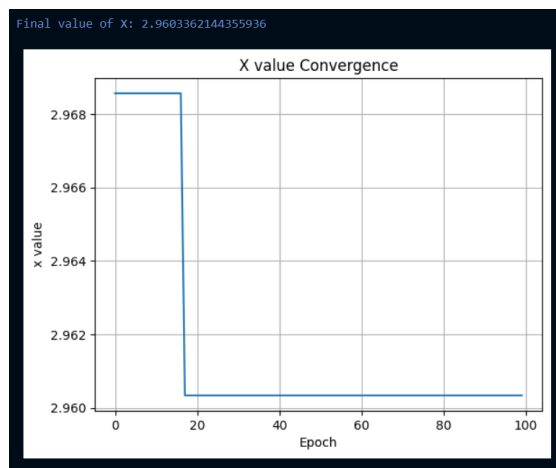


Figura 2. Convergencia con velocidad x

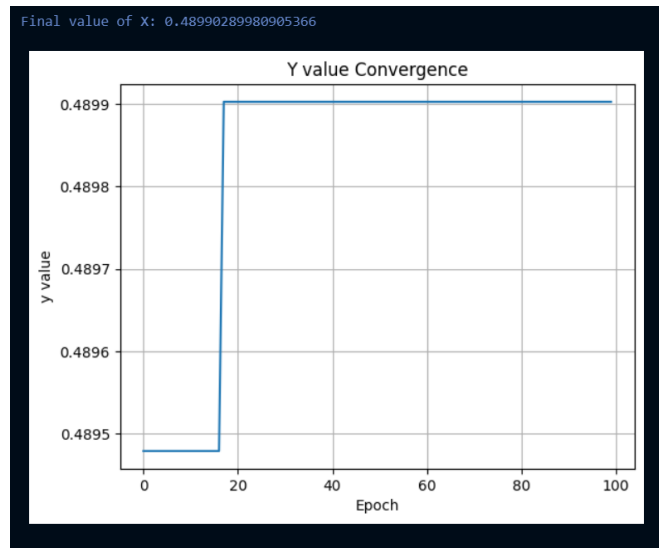


Figura 2. Convergencia y con velocidad

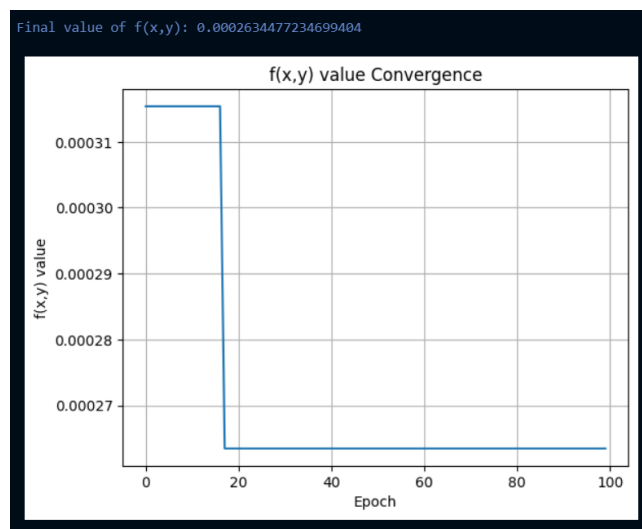


Figura 3. Convergencia $f(x,y)$ con velocidad

Código:

https://github.com/Gabeinstein/artificial_intelligence_fall23_gona/blob/main/genetic%20algorithm/genetic%20algorithm.ipynb

