Introducción

El algoritmo de Optimización por Enjambre de Partículas (PSO) es una técnica de optimización inspirada en el comportamiento social de los pájaros o peces. En PSO, un grupo de partículas (o agentes) se mueven en el espacio de búsqueda para encontrar la solución óptima a un problema dado. Cada partícula representa una posible solución al problema y se mueve en función de su propia experiencia y la de las mejores partículas en el enjambre.

PSO (Particle Swarm Optimization) es una metaheurística en el campo de los sistemas inteligentes que permite encontrar el valor óptimo de una función objetivo determinada. Para ello utiliza un conjunto de partículas que se desplazan y colaboran entre sí en un espacio n-dimensional. Este proceso demanda un costo computacional que depende de la función analizada y de la estrategia empleada

Lógica principal PSO

**(w \* particle.velocity)**: Es un valor que determina cuánta inercia se mantiene. Si es alto, las partículas tienden a mantener su dirección actual; si es bajo, son más propensas a explorar nuevas áreas.

**(c1 \* r1 \* (particle.best\_position - particle.position)) componente cognitiva:**

c1 es el peso cognitivo, que determina cuánto la partícula se guía por su mejor posición conocida. r1 son números aleatorios entre 0 y 1 que ayudan a introducir aleatoriedad y exploración.

**Componente social (c2 \* r2 \* (self.global\_best\_position - particle.position))** c2 es el peso social, que determina cuánto la partícula se guía por la mejor posición conocida por todo el enjambre. r2 son números aleatorios entre 0 y 1 que introducen aleatoriedad.

**r1 y r2** introducen aleatoriedad en las actualizaciones de velocidad. Esta aleatoriedad ayuda al algoritmo a explorar el espacio de búsqueda de manera más efectiva y evita quedar atrapado en mínimos locales. La aleatoriedad es clave para que el PSO pueda explorar y explotar el espacio de búsqueda de manera equilibrada.

**DExPSO** En DExPSO, las partículas actualizan sus velocidades en función de una distribución de doble exponencial. Esta distribución ayuda a controlar el equilibrio entre la exploración y la explotación en el espacio de búsqueda, mejorando la capacidad del algoritmo para escapar de los óptimos locales y explorar nuevas regiones.”

“La idea clave aquí es aproximar la variable aleatoria uniforme mediante otra variable aleatoria con soporte en toda la recta real, como en un proceso de suavizado. Estos requisitos garantizan características deseables, como que las propiedades de convergencia del algoritmo se transferirán al nuevo algoritmo

PSO 2 dimensiones : En su versión bidimensional, PSO busca encontrar la solución óptima de un problema considerando dos variables. Imagina un enjambre de partículas en un espacio de búsqueda, donde cada partícula representa una posible solución. Estas partículas se mueven hacia regiones más prometedoras del espacio según su posición y velocidad

La implementación en 2 dimensiones expande las capacidades de PSO para abordar problemas más complejos, permitiendo la exploración simultánea de dos variables y mejorando la convergencia hacia soluciones óptimas en el espacio de búsqueda bidimensional.