



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto tecnológico de Culiacán

Actividad:

Algoritmos Basados en Enjambre de Partículas

Alumnos:

Ivan Eduardo Ramírez moreno

Carrera:

Ing. en sistemas computacionales

Docente:

ZURIEL DATHAN MORA FELIX

Materia:

Tópicos de IA

Semestre:

10

Algoritmos Basados en Enjambre de Partículas

Introducción

En el mundo moderno, caracterizado por una creciente complejidad en los sistemas y problemas a resolver, los algoritmos inspirados en la naturaleza han emergido como soluciones eficaces y adaptables. Estos algoritmos toman como modelo comportamientos observados en la biología, la física y otras ciencias naturales, para resolver problemas computacionales que, de otra forma, serían intratables.

Uno de los algoritmos más representativos de esta corriente es el **Algoritmo de Optimización por Enjambre de Partículas (PSO)**, una técnica inspirada en el comportamiento social de especies animales como aves y peces, particularmente su capacidad de moverse en grupo de manera eficiente para alcanzar objetivos como buscar comida o evitar depredadores. Su eficacia ha sido demostrada en una gran variedad de problemas, desde funciones matemáticas hasta aplicaciones industriales y de ingeniería.

Antecedentes Históricos

El PSO fue desarrollado en 1995 por **James Kennedy** (psicólogo social) y **Russell Eberhart** (ingeniero eléctrico), quienes inicialmente estudiaban el comportamiento de las aves durante el proceso de búsqueda de alimento. Su objetivo era modelar un sistema de auto organización capaz de encontrar soluciones óptimas de forma colectiva, sin necesidad de una supervisión centralizada.

Este enfoque pronto se convirtió en una técnica de optimización robusta, ubicándose junto a algoritmos evolutivos como los **algoritmos genéticos** y las **estrategias evolutivas**, pero con un menor número de parámetros y una implementación más sencilla.

Fundamentos Biológicos del PSO

El principio fundamental detrás del PSO es la **inteligencia de enjambre**, un fenómeno en el que agentes simples, actuando localmente y sin una autoridad central, generan comportamientos complejos e inteligentes como grupo.

Ejemplo natural: bandadas de aves y cardúmenes de peces

Estos grupos animales muestran tres comportamientos clave que fueron abstraídos al PSO:

1. **Separación:** Evitan colisiones con sus vecinos.
2. **Alineación:** Ajustan su velocidad para coincidir con la del grupo.
3. **Cohesión:** Se mueven hacia el centro del grupo más cercano.

Estos comportamientos permiten a las especies explorar grandes áreas, encontrar rutas óptimas y adaptarse rápidamente a cambios en el entorno, sin planificación previa ni comunicación explícita compleja.

Estructura del Algoritmo PSO

El PSO trabaja con un **conjunto de partículas** que representan posibles soluciones dentro del espacio de búsqueda. Cada partícula se mueve influida por:

- Su experiencia individual (la mejor solución que ha encontrado).
- La experiencia colectiva del grupo (la mejor solución encontrada por cualquier partícula).

A lo largo de varias iteraciones, las partículas se ajustan a través de dos fases:

- **Exploración:** Búsqueda amplia de soluciones en el espacio del problema.
- **Explotación:** Refinamiento de las mejores soluciones conocidas.

Componentes clave

- **Posición de la partícula:** Representa una posible solución.
- **Velocidad:** Define la dirección y magnitud del movimiento.
- **Mejor posición personal (pbest):** Mejor solución encontrada por esa partícula.
- **Mejor posición global (gbest):** Mejor solución encontrada por el enjambre.

Formulación Matemática

Actualización de velocidad

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (p_i^{best} - x_i(t)) + c_2 \cdot r_2 \cdot (g^{best} - x_i(t))$$

Actualización de posición

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

Donde:

- x_i : Posición de la partícula i .
- v_i : Velocidad de la partícula.
- p_i^{best} : Mejor posición encontrada por la partícula.
- g^{best} : Mejor posición encontrada por el enjambre.
- c_1, c_2 : Pesos de los componentes cognitivo y social.
- w : Factor de inercia (controla el equilibrio entre exploración y explotación).
- r_1, r_2 : Números aleatorios en el intervalo $[0, 1]$.

Comparación con Otros Algoritmos Evolutivos

Característica	PSO	Algoritmos Genéticos (GA)	Búsqueda Tabú / Hill Climbing
Inspiración	Comportamiento social	Evolución biológica	Caminos de solución secuencial
Operadores principales	Velocidad y posición	Cruza, mutación, selección	Memoria / evaluación local
Estructura de solución	Partículas	Cadenas de genes	Una única solución activa
Convergencia	Rápida (pero puede estancarse)	Lenta, pero robusta	Muy local
Diversidad poblacional	Media	Alta	Muy baja

Aplicaciones del PSO

El PSO ha sido exitosamente aplicado a diversos campos:

Ingeniería

- Optimización de diseños mecánicos y estructurales.
- Ajuste de parámetros de controladores PID.
- Diseño de circuitos eléctricos.

Inteligencia Artificial

- Entrenamiento de redes neuronales.
- Ajuste de hiperparámetros en algoritmos de machine learning.

Robótica

- Planificación de trayectorias para robots autónomos.
- Coordinación de múltiples drones o vehículos.

Ciencias Naturales

- Modelado de procesos biológicos y ecológicos.
- Predicción de fenómenos climáticos.

Finanzas y Logística

- Predicción de mercados financieros.
- Optimización de rutas de entrega y cadenas de suministro.

Ventajas y Desventajas

Ventajas:

- Fácil de implementar.
- Convergencia rápida en muchos problemas.
- No requiere derivadas ni supuestos fuertes sobre el problema.

Desventajas:

- Puede quedar atrapado en óptimos locales.
- Su rendimiento depende mucho del ajuste de parámetros.
- No es el más eficiente para todos los tipos de problemas (por ejemplo, discretos).

Variantes del PSO

- **PSO con Inercia Variable:** Ajusta dinámicamente el valor de w para favorecer exploración al principio y explotación al final.
- **PSO Híbrido:** Se combina con algoritmos como búsqueda local, algoritmos genéticos o recocido simulado.
- **PSO Multiobjetivo (MOPSO):** Permite trabajar con más de una función objetivo, usando estrategias de pareto para seleccionar soluciones óptimas.
- **PSO Cuántico:** Introduce conceptos de la computación cuántica para mejorar la exploración del espacio de búsqueda.

Ejemplo Práctico: Optimización de una Función Matemática

Supongamos que deseamos minimizar la función:

$$f(x, y) = x^2 + y^2$$

Este problema tiene una solución óptima global en el punto (0,0). El PSO puede encontrar esta solución generando partículas en el plano y actualizando sus posiciones iterativamente hasta que converjan al mínimo.

Al graficar la trayectoria de las partículas, se puede observar cómo inicialmente exploran el espacio y gradualmente se acercan al óptimo.

Perspectivas Futuras:

El PSO continúa evolucionando. Algunas líneas de investigación actuales incluyen:

- PSO distribuido en entornos paralelos y en la nube.
- PSO aplicado a problemas dinámicos, donde el objetivo cambia con el tiempo.
- PSO asistido por aprendizaje automático, donde redes neuronales predicen regiones prometedoras.
- Uso en criptografía, computación cuántica y bioinformática.

El creciente interés por la computación natural sugiere que los algoritmos como PSO seguirán siendo fundamentales en la resolución de problemas reales y complejos en diversas áreas.

Conclusión

El Algoritmo de Optimización por Enjambre de Partículas es un ejemplo brillante de cómo la observación de la naturaleza puede traducirse en soluciones efectivas para desafíos computacionales. Su diseño, inspirado en la cooperación animal, permite encontrar soluciones óptimas de manera eficiente, descentralizada y robusta. A medida que las tecnologías evolucionan, el PSO y sus variantes prometen seguir ampliando sus fronteras de aplicación, demostrando que la inteligencia colectiva no es exclusiva de los seres vivos, sino también de las máquinas que aprenden de ellos.

Bibliografía

Clerc, M. (1999). *The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization*. Proceedings of the 1999 Congress on Evolutionary Computation (CEC99), 1951–1957. <https://doi.org/10.1109/CEC.1999.785513>

Kennedy, J., & Eberhart, R. (1995). *Particle swarm optimization*. Proceedings of ICNN'95 - International Conference on Neural Networks, 1942–1948. <https://doi.org/10.1109/ICNN.1995.488968>

Seeley, T. D. (2010). *Honeybee democracy*. Princeton University Press.

Shi, Y., & Eberhart, R. C. (1998). *A modified particle swarm optimizer*. 1998 IEEE International Conference on Evolutionary Computation Proceedings, 69–73. <https://doi.org/10.1109/ICEC.1998.699146>

Tan, Y., & Ding, Y. (2008). *A survey on particle swarm optimization algorithm*. Evolutionary Computation, 1(1), 1–10.