

Metaheurísticas

CONFERENCIA #5. INTELIGENCIA DE ENJAMBRES

Swarm Intelligence

- ▶ Un enjambre (swarm) puede ser definido como una colección estructurada de organismos (agentes) que interactúan.
- ▶ La inteligencia no está en los individuos sino en el colectivo.

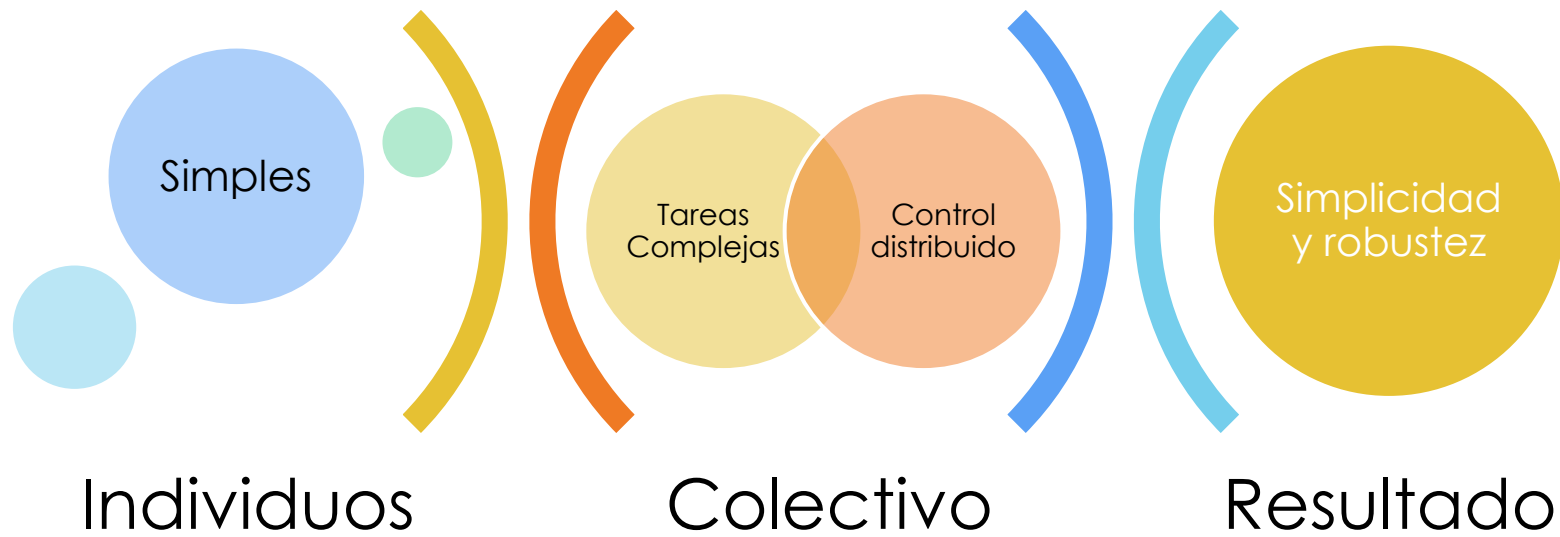
Swarm Intelligence

- ▶ Swarm Intelligence (SI) es la propiedad de un sistema por la cual el comportamiento colectivo de agentes (no sofisticados) interactúa localmente con el entorno proporcionando un patrón global de funcionamiento coherente como “emergente”.
- ▶ SI proporciona una base con la cual es posible explorar la resolución de problemas colectivamente (o de forma distribuida) sin un control centralizado ni un model global de comportamiento.

Swarm Intelligence

- ▶ Se define como:
- ▶ La inteligencia colectiva que emerge de un grupo entidades simples, llamadas usualmente agentes.
- ▶ Cualquier intento de diseñar algoritmos inspirados por el comportamiento colectivo de colonias de insectos sociales u otras sociedades de animales.

Swarm Intelligence



Naturaleza Inteligente - Bandas, Manada y Cardume

- ▶ Banco de peces
 - ▶ Breder en 1954 describió matemáticamente la dependencia de atracción que ejerce el cardumen sobre cada pez
 - ▶ D.Hamilton en 1971 – “Geometría de la manada”
 - ▶ El animal en el vértice de la manada es más propenso a ser capturado por un depredador

Naturaleza Inteligente - Bandas, Manada y Cardume

- ▶ Banda de pájaros
 - ▶ Craig Reynolds, 1987: Boids
 - ▶ Simulación del vuelo de un pájaro
 - ▶ Tres fuerzas locales:
 - ▶ Evitar COLISION
 - ▶ Fuerza para evitar los choques mutuos
 - ▶ VELOCIDAD DE ENCUENTRO
 - ▶ Tratan de mantener la misma velocidad de sus vecinos en la banda
 - ▶ VUELO CENTRADO
 - ▶ Trata de moverse hacia el centro de la banda

Ejemplos de Swarm Intelligence

- ▶ Colonias de hormigas (Ant colonies)
- ▶ Bandas de pájaros (Bird flocks)
- ▶ Bancos de peces (Fish schools)
- ▶ Ant Colony Optimization (ACO)
- ▶ Artificial Bee Colony (ABC)

- ▶ Modelos computacional a estudiar:
 - ▶ Particle Swarm Optimization (PSO)

Stigmergy

- ▶ Interacción social directa o indirecta
- ▶ La interacción indirecta ocurre cuando un individuo cambia el ambiente y el otro agente responde al nuevo ambiente
- ▶ Este tipo de comunicación se conoce como *stigmergy*

Particle Swarm Optimization



Particle Swarm Optimization

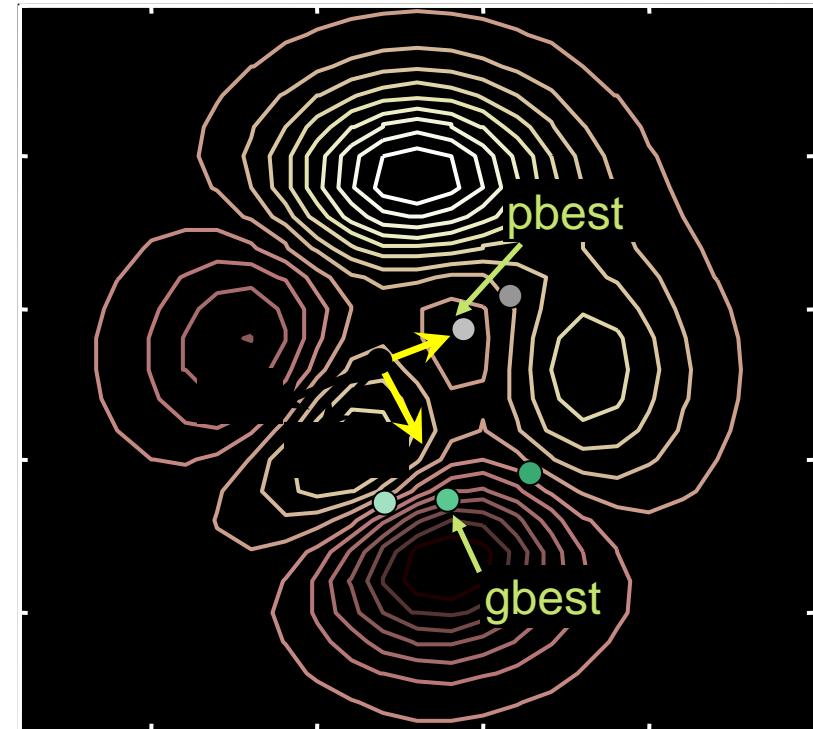
- ▶ Técnica de optimización estocástica basada en una población
- ▶ Creada por Eberhart and Kennedy en 1995

Vuelo del pájaro

- ▶ Cada pájaro (partícula) es tratado como un punto en un espacio N dimensional el cual ajusta su propio “vuelo” de acuerdo a su propia experiencia y la experiencia del resto de la banda
- ▶ La banda (swarm) vuela por el espacio buscando regiones prometedoras

Particle Swarm Optimization

- ▶ Las Partículas pueden ser simples agentes que vuelan a través del espacio de búsqueda y almacenan (y posible-mente comunican) la mejor solución que han descubierto
- ▶ En PSO, las partículas nunca mueren
- ▶ La pregunta es, ¿Cómo se mueve una partícula desde una localización a otro del espacio de búsqueda?



Principios básicos de PSO

- ▶ Cada partícula tiene una posición y velocidad en el espacio de búsqueda
 - ▶ La Posición define el contenido de la solución candidata
- ▶ Cada partícula tiene una medida de calidad – parecido a la “fitness” de un individuo en los algoritmos evolutivos
 - ▶ Búsqueda heurística
- ▶ Cada partícula (individuo) puede interactuar con un numero de vecinos
 - ▶ Cada partícula conoce el valor (*fitness*) y posición de cada uno de sus vecinos
 - ▶ Implementa los principios de “comparar” e “imitar”
- ▶ Cada partícula “aprende” ajustando su posición y velocidad:
 - ▶ Parcialmente atraído a su mejor posición (*best solution*) hasta el momento
 - ▶ Parcialmente atraído a la mejor posición de la vecindad

Particle Swarm Optimization

- ▶ Representación de un problema de optimización: **Minimize $f=x^2+y^2$.**
- ▶ Cada solución candidata de un problema de optimización continuo se describe (codifica) por un vector de números reales: (x,y)
- ▶ Cada solución candidata se denomina PARTICULA y representa un individuo de una población
- ▶ La población es un conjunto de vectores y es llamada ENJAMBRE (SWARM)
- ▶ Las partículas cambian sus componentes y se mueven (vuelan) en un espacio R^n
- ▶ Ellas pueden evaluar su posición actual usando la función a ser optimizada
- ▶ En este caso la función es f
- ▶ Las Partículas también se comparan con sus vecinas e imitan a la mejor de esa vecindad

Particle Swarm Optimization

- ▶ Comparte muchas similitudes con otras técnicas de computación evolutiva como los AG
- ▶ Pero no usa operadores como mutación y cruce
- ▶ Más fácil de implementar
- ▶ Pocos parámetros que ajustar

PSO como una técnica evolutiva

- ▶ Se inicializa con una población de “soluciones aleatorias”
- ▶ Se busca el óptimo actualizando generaciones
- ▶ La reproducción se basa en las viejas generaciones

Particle Swarm Optimization

- ▶ Aplicaciones exitosas en:
 - ▶ Optimización de funciones y multiobjetivo
 - ▶ Entrenamiento de RNA
 - ▶ Control de sistemas borrosos
 - ▶ Problemas de satisfacción de restricciones
 - ▶ Etc...

PSO: Algoritmo

- ▶ La bandada virtual vuela a través del espacio de problema N dimensional, evalúa las diferentes posiciones que alcanza cada partícula según la función a optimizar llevando un registro de los mejores puntos alcanzados

PSO: Componentes

- ▶ X_i : Partícula (vector de N dimensiones)
- ▶ $\{X_1, X_2, \dots\}$: Bandada (conjunto de partículas)
- ▶ $\{V_1, V_2, \dots\}$: Velocidades (vectores N dimensional asociado a cada partícula que indica su movimiento)
- ▶ $\{X_{pbest1}, X_{pbest2}, \dots\}$: Mejores puntos del espacio localizados por cada partícula
- ▶ X_{gbest} : Mejor punto localizado por la bandada

Inicialización de la Nube de Partículas

- ▶ La nube se inicializa generando las posiciones y las velocidades iniciales de las partículas
- ▶ Las posiciones se pueden generar aleatoriamente en el espacio de búsqueda, de forma regular, o con una combinación de ambas
- ▶ Las velocidades se generan aleatoriamente, con cada componente en el intervalo $[-V_{\max}, V_{\max}]$
 - ▶ No es conveniente fijarlas a cero, no se obtienen buenos resultados
 - ▶ V_{\max} será la velocidad máxima que pueda tomar una partícula en cada movimiento

Movimiento de las Partículas

- ▶ La pregunta es: ¿Cómo se mueve una partícula de una posición del espacio de búsqueda a otra?
- ▶ Se hace simplemente añadiendo el vector velocidad V_i al vector posición X_i para obtener un nuevo vector posición:

$$X_i \leftarrow X_i + V_i$$

- ▶ Una vez calculada la nueva posición de la partícula, se evalúa ésta. Si el nuevo fitness es mejor que el que la partícula tenía hasta ahora, $pBest_fitness$, entonces:

$$pBest_i \leftarrow X_i \quad ; \quad pBest_fitness \leftarrow x_fitness.$$

PSO: Reglas de actualización

$$V_i = w * V_i + c1 * rand() * (Xpbest_i - X_i) + c2 * rand() * (Xgbest - X_i)$$

$$X_i = X_i + V_i$$

Rojo: memoria de la partícula

Azul: parte cognitiva (conocimiento privado)

Verde: parte social (permite la colaboración)

Algoritmo PSO

Inicializar aleatoriamente la población de X_i y V_i , X_{pbesti} de cada partícula con copia de X_i , X_{gbest} con el mejor valor.

Repeat

For each X_i

Calcular $V_i(t+1)$ y limitarla a $[-V_{max}, +V_{max}]$

Actualizar X_i

endFor

For each X_i

Evaluar X_i

Actualizar X_{pbesti}

endFor

Actualizar X_{gbest}

Until condición de terminación

PSO: parámetros

- ▶ Número de partículas (tamaño de la bandada)
- ▶ Número de generaciones
- ▶ Peso de la inercia (w)
- ▶ Razón de aprendizaje cognitivo ($c1$)
- ▶ Razón de aprendizaje social ($c2$)

PSO: valores recomendados para los parámetros

- ▶ Número de partículas (tamaño de la bandada): entre 10 y 40
- ▶ Número de generaciones: 100-200
- ▶ A mayor valor de ambos parámetros más oportunidad de encontrar el óptimo pero MAYOR COSTO COMPUTACIONAL

PSO: valores recomendados para los parámetros

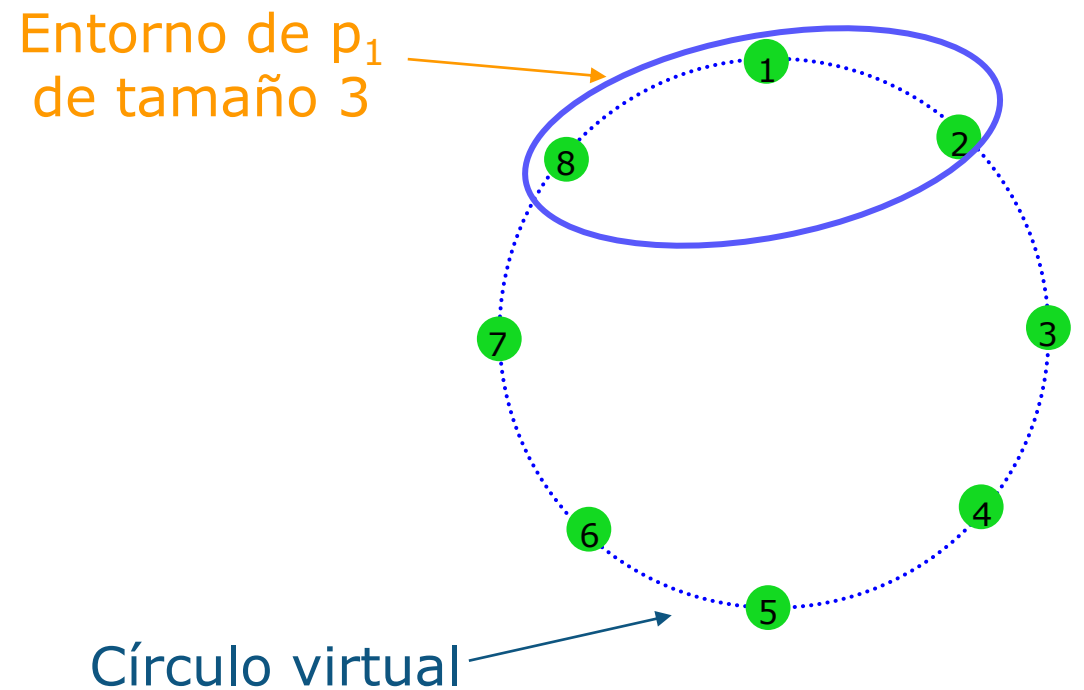
- ▶ Razón de aprendizaje cognitivo ($c1$) y
- ▶ Razón de aprendizaje social ($c2$):
- ▶ $c1=c2=1.5$ $c1=c2=2$
- ▶ Valores bajos permiten explorar regiones diversas antes de dirigirse al objetivo, mientras que valores mayores permiten saltos bruscos a este
 - ▶ Sugerencia $\text{rand1}()c1 + \text{rand2}()*c2 \leq 4$

Topología de la Nube de Partículas

- ▶ Las topologías definen el entorno de cada partícula individual. La propia partícula siempre pertenece a su entorno
- ▶ Los entornos pueden ser de dos tipos:
 - ▶ Geográficos: se calcula la distancia de la partícula actual al resto y se toman las más cercanas para componer su entorno
 - ▶ Sociales: se define a priori una lista de vecinas para partícula, independientemente de su posición en el espacio
- ▶ Los entornos sociales son los más empleados
- ▶ Una vez decidido el entorno, es necesario definir su tamaño. El algoritmo no es muy sensible a este parámetro (3 o 5 son valores habituales con buen comportamiento)

Topología de la Nube de Partículas

- La topología social más empleada es la de anillo, en la que se considera un vecindario circular
- Se numera cada partícula, se construye un círculo virtual con estos números y se define el entorno de una partícula con sus vecinas en el círculo:



Topología de la Nube de Partículas

- ▶ Cuando el tamaño es toda la nube de partículas, el entorno es a la vez geográfico y social, y tenemos la PSO global
- ▶ PSO Global vs. PSO Local: La versión global converge más rápido pero cae más fácilmente en óptimos locales y viceversa

Consideraciones finales PSO

- ▶ Es más rápido en localizar la solución óptima que los AG
- ▶ Tiene potente capacidad de exploración
- ▶ Su proceso de búsqueda gradual aproxima las soluciones óptimas
- ▶ Se basa en conceptos muy simples
- ▶ Poco costo computacional en términos de memoria y tiempo

Conclusiones

- ▶ No hay un algoritmo que satisfaga todos los problemas (Teorema *No Free Lunch*)
- ▶ Los algoritmos de Inteligencia Colectiva son fáciles de implementar, tienen poco costo computacional y generan buenas soluciones



Preguntas

Bibliografía

- ▶ Libros en la carpeta de la asignatura
- ▶ Otros disponibles en Internet