INTELIGENCIA ARTIFICIAL (1INF24)

UNIDAD 1: Introducción a la IA. Búsqueda y optimización en IA

Tema 3: Algoritmos heurísticos, metaheurísticos y bioinspirados (Parte 2)

Dr. Edwin Villanueva Talavera

Contenido

- Inteligencia Colectiva
- Algoritmos de Inteligencia Colectiva
  - Particle Swarm Optimization (PSO)
  - Artificial Bee Colony (ABC)

Inteligencia Colectiva (IC)

Pepinos de mar

Medusas

Carabelas portuguesas

Es una colonia de zooides que al nacer se pegan unos a otros y siguen juntos de por vida.
Cada uno de los organismos que conforman la carabela se especializan en una función de supervivencia, como: defensa, reproducción o alimentación.

A falta de cerebro, la unión hace la fuerza.

Inteligencia Colectiva (IC)

- Se define IC como el aprendizaje que se efectúa a nivel de grupo. La base de este tipo de inteligencia es la experiencia compartida.
- IC (o Swarm intelligence)[1] fue introducida por Gerardo Beni, Suzanne Hackwood y Jing Wang en 1989, cuando investigaban las propiedades de agentes auto-organizados para sistemas robóticos celulares. Se inspiraron en el comportamiento social de colonias de insectos, como las hormigas.

Las habilidades de estos sistemas sociales parecen trascender las habilidades individuales.

Capacidades y fortalezas grupales

Se deben a un pequeño grupo de simples interacciones de bajo-nivel entre individuos y su entorno

[1] E. Bonabeau, et. al , 1999. Swarm Intelligence. From Nature to Artificial Systems. Oxford University Press.

Inteligencia Colectiva (IC)

□ Particle Swarm Optimization (PSO)

- Conjunto de técnicas inspiradas en el comportamiento social de bandadas de aves, bancos de peces y otros.

bandadas de aves

cardúmenes de peces

Inteligencia Colectiva (IC)

- Artificial Bee Colony (ABC)
  - Conjunto de técnicas inspiradas por las actividades de una colonia de abejas.

Inteligencia Colectiva (IC)

- Ant Colony Optimization (ACO)
  - Conjunto de técnicas inspiradas por las actividades de una colonia de hormigas

Colonias de hormigas

Inteligencia Colectiva (IC)

Propiedades Generales:

- Algoritmos estocásticos, basados en poblaciones y usan representación en punto flotante.
- Fueron diseñados intentando solucionar problemas de optimización desafiantes.
- Estos algoritmos controlan pocos parámetros (fáciles de usar)
- Tienen buenas propiedades de convergencia.

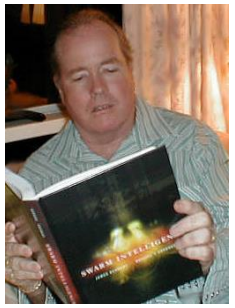
Particle Swarm Optimization (PSO)



Particle Swarm Optimization (PSO)

- En 1995, PSO fue desarrollado por James Kennedy y Russell Eberhart[2], para resolver problemas de optimización continua

James Kennedy



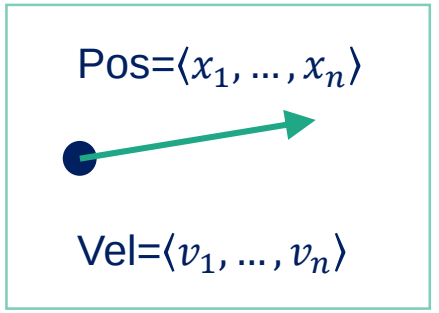
Russell Eberhart

- Su trabajo se inspiro en el comportamiento y dinámica de los movimientos de los pájaros, insectos y peces.
- Para crear PSO usaron las reglas de cohesión y alineamiento del modelo de Reynolds.
- PSO, al igual que AG es un método basado en poblaciones , pero diferentemente de los AG, la metáfora por detrás es la cooperación en lugar de la rivalidad. En PSO, los miembros del enjambre nunca mueren.

[2] Kennedy, J.; Eberhart, R. (1995). Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks Vol. IV: 1942–1948.

Particle Swarm Optimization (PSO)

- **Una población en PSO se compone de un conjunto de partículas** que representan las soluciones a un problema.
- En cada iteración, todas las partículas se mueven por el espacio del problema con el objetivo de encontrar una solución optima global.
- Una partícula en PSO tiene:
  - Un vector de su posición actual.
  - Un vector de velocidad que dirige su movimiento.
  - La capacidad de intercambiar información con sus vecinos.
  - La capacidad de memorizar una posición anterior.



Particle Swarm Optimization (PSO)

Movimiento de partículas:

Una partícula **i** en la interacción k+1 se mueve a una posición x_i^{k+1} que depende de su posición anterior (x_i^k) y su nueva velocidad (v_i^{k+1}):

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (1)$$

Aprendizaje Cognitivo — Aprendizaje Social

$$v_i^{k+1} = \omega . v_i^k + \varphi_1 . rnd_1 . (pBest_i - x_i^k) + \varphi_2 . rnd_2 . (g - x_i^k) \quad (2)$$

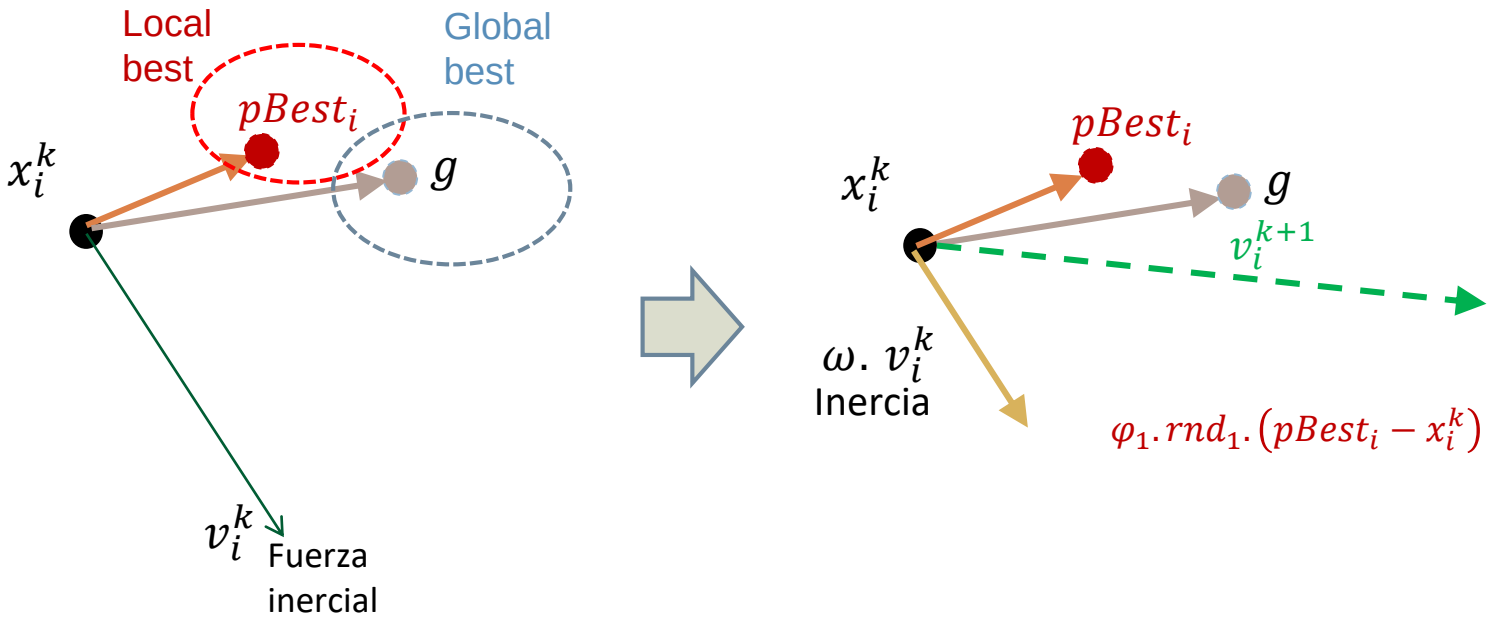
- $\omega . v_i^k$: Fuerza inercial
- $\varphi_1 . rnd_1 . (pBest_i - x_i^k)$: Fuerza hacia la mejor posición local $pBest_i$
- $\varphi_2 . rnd_2 . (g - x_i^k)$: Fuerza hacia la mejor posición global g del enjambre.

Donde:

- v_i^k, velocidad anterior
- ω representa el factor inercia
- $pBest_i$,mejor posición encontrada por particula i
- g , mejor posición global ya vista por el swarm
- φ_1 y φ_2, pesos de tendencia individual o social respectivamente.
- rnd_1 y rnd_2 , son números aleatorios entre $\{0,1\}$.

Particle Swarm Optimization (PSO)

Movimiento de partículas:



Aprendizaje Cognitivo — Aprendizaje Social

$$v_i^{k+1} = \omega . v_i^k + \varphi_1 . rnd_1 . (pBest_i - x_i^k) + \varphi_2 . rnd_2 . (g - x_i^k)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}$$

Particle Swarm Optimization (PSO)

Algorithm 2 Pseudocodigo del algoritmo PSO Canónico

1: $S \leftarrow SwarmInitialization()$
2: **while** not stop condition **do**
3: **for** each particle x_i of the swarm S **do**
4: evaluate(x_i)
5: **if** $fitness(x_i)$ is better than $fitness(pBest_i)$ **then**
6: $pBest_i \leftarrow x_i$
7: **end if**
8: **if** $fitness(pBest_i)$ is better than $fitness(g)$ **then**
9: $g \leftarrow pBest_i$
10: **end if**
11: **end for**
12: **for** each particle x_i of the swarm S **do**
13: $v_i^{k+1} = \omega \cdot v_i^k + \varphi_1 \cdot rnd_1 \cdot (pBest_i - x_i^k) + \varphi_2 \cdot rnd_2 \cdot (g - x_i^k)$
14: $x_i^{k+1} = x_i^k + v_i^{k+1}$
15: **end for**
16: **end while**
17: **Output:** best solution found

Particle Swarm Optimization (PSO)

Elección de parámetros:

$$v_i^{k+1} = \omega.v_i^k + \varphi_1.rnd_1.\left(pBest_i - x_i^k\right) + \varphi_2.rnd_2.\left(g - x_i^k\right)$$

- Algunas recomendaciones de valores para:
  - Inercia de la partícula a ω =0.5, porque debe influir menos.
  - Para φ_1 y φ_2= 2 (tendencia local φ_1 y tendencia global del enjambre φ_2).
- Una mejora del método consiste en empezar con una inercia relativamente alta (ω = 1.4) que se vaya reduciendo en cada iteración, por ejemplo, multiplicándola por un factor r <1 para estabilizar el sistema.
- Otras propuestas:

$$\text{Inertia weight } \omega = (\omega_1 - \omega_2)\left(\frac{\text{maxiter} - \text{iter}}{\text{iter}}\right) + \omega_2,$$

$$\text{Inertia weight } \omega_i = \left(1.1 - \frac{g\text{best}_i}{(p\text{best}_i)}\right).$$

Hacer decaer linealmente en el tiempo la inercia

Particle Swarm Optimization (PSO)

Consideraciones:

- Población inicial: crea un conjunto S de partículas, donde a cada partícula x_i se le asigna una posición inicial y una velocidad, ambas deben estar acotadas por límites:

$$x_i \leftarrow \cup\ (inf, sup) \qquad v_i \leftarrow \cup\ (-|sup - inf|, |sup - inf|)$$

- Iteraciones:
  - Tras actualizar todas las partículas, se comparan sus mejores posiciones $pBest$ con la mejor posición global g, si algún $pBest$ es mejor que g, entonces: $g \leftarrow pBest_i$.
  - Al acabar las iteraciones, g contiene la mejor solución encontrada por el algoritmo

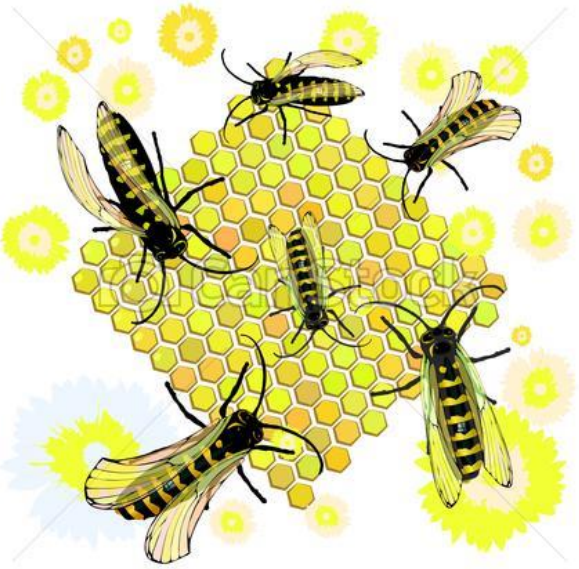
Particle Swarm Optimization (PSO)

Consideraciones:

- Los algoritmos tipo PSO vienen siendo aplicados en problemas de optimización combinatoria, donde la dimensión del espacio completo de soluciones es exponencial (NP-hard) con respecto a la dimensión de la representación del problema.
- A pesar de que PSO inicialmente se formulo para trabajar con representación real, están surgiendo nuevas versiones que permiten trabajar con representaciones discretas.
  - Un ejemplo es la versión de PSO basada en espacios geométricos (*Geometric PSO*)[3]. Las partículas se influyen dependiendo de su distancia en el espacio. El algoritmo se modifica para trabajar con soluciones binarias utilizando la distancia de Hamming.

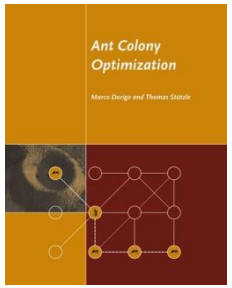
[3] A. Moraglio, C. Di Chio, and R. Poli. Geometric Particle Swarm Optimization. In 10th European conference on Genetic Programming (EuroGP2007), volume 4445 of Lecture Notes in Computer Science. Springer, Abril 2007.

Artificial Bee Colony (ABC)



Artificial Bee Colony (ABC)

- El algoritmo de colonias de abejas o *Artificial Bee Colony* (ABC) [6], fue presentado por Dervis Karaboga en 2005.



- ABC es inspirado en el comportamiento inteligente observado en las abejas domesticas.
- ABC es un algoritmo propuesto para resolver problemas de optimización. Está basado **en poblaciones** en el cual **los individuos, llamados fuentes de comida**, son modificados por las abejas artificiales.
  - El objetivo de estas abejas es descubrir los lugares de comida con néctar, hasta finalmente encontrar el lugar con mayor cantidad de néctar (optimo global).

[6] D. Karaboga. An idea based on honey bee swarm for numerical optimization. Tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

Artificial Bee Colony (ABC)

- En un sistema ABC, las abejas artificiales se mueven por el espacio de búsqueda eligiendo fuentes de néctar dependiendo de su experiencia pasada y de su compañera de colmena.

Algunas abejas se mueven aleatoriamente sin influencia externa (exploradoras).

Cuando encuentran una fuente de néctar mayor, memorizan su posición y olvidan la anterior.

- De este modo, ABC combina métodos de búsqueda local y búsqueda global, intentando equilibrar el balance entre exploración y explotación del algoritmo.

Artificial Bee Colony (ABC)

Modelo Biológico:

El modelo biológico que guía la búsqueda de néctar es un proceso de optimización y consta de los siguientes elementos:

1) **Fuente de alimento:** Es un valor numérico que indica su potencial.
2) **Abejas empleadas (recolectoras):** Exploran una fuente de alimento. Comunican su ubicación y rentabilidad, con una cierta probabilidad, a sus demás compañeras observadoras.
3) **Abejas desempleadas:** Están en constante búsqueda de una fuente de alimento. Hay dos tipos:

   Exploradoras (Scout): se encargan de buscar en el entorno que rodea a la colmena nuevas fuentes de alimento.

   Observadoras (onlooker bee): permanecen en la colmena para elegir alguna de las fuentes de alimento.

Artificial Bee Colony (ABC)

Algoritmo Básico:

- Población inicial: es iniciada con un número SN de vectores de valores reales n-dimensionales, generados de la siguiente manera:

$$x_i = x_{min} + rand(0,1).(x_{max} - x_{min}) \quad (1)$$

- Fase de búsqueda por las abejas empleadas:

Las abejas empleadas representan operadores de variación. Cuando visitan fuentes de alimento, calculan una nueva posición $v_{i,g}$, así:

$$v_{i,g} = x_i + r.(x_i - x_k) \quad (2)$$

donde:

- x_i, fuente de alimento donde se encuentra la abeja en ese momento.
- x_k, fuente de alimento seleccionada aleatoriamente y diferente de x_i.
- g es el ciclo actual y
- r es un número real aleatorio entre [−1, 1].

Artificial Bee Colony (ABC)

Algoritmo Básico:

- Una vez que se obtiene $v_{i,g}$, se evaluará y se comparará con x_i, luego se selecciona la mejor dependiendo de los valores de aptitud (*fitness*) que representan. Si la aptitud de $v_{i,g}$ $\text{es mejor } x_i$, entonces $v_{i,g}$ reemplazará x_i , es decir $x_i \leftarrow v_{i,g}$, y se convertirá en un nuevo miembro de la población; de lo contrario x_i se conserva.

Artificial Bee Colony (ABC)

Algoritmo Básico:

- Fase de selección por las abejas observadoras (onlooker bee)

  Las abejas observadoras seleccionan las fuentes de alimento de acuerdo a una probabilidad p_i asociada a la fuente de alimento y este valor es proporcional a la cantidad de néctar que tiene la solución. Se calcula de la siguiente manera:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \qquad (3)$$

  donde:

  - fit_i es el valor de la aptitud (fitness) de la solución i,
  - SN es el número de fuentes de alimento.

Nota: El esquema de selección puede ser: ruleta, muestreo estocástico universal, torneos u otro esquema de selección.

- Después de seleccionar la fuente de alimento, las abejas observadoras crearán una nueva posición candidata en el vecindario de la fuente de alimento seleccionado utilizando (2)

Artificial Bee Colony (ABC)

Algoritmo Básico:

- Después de que todas las abejas empleadas y observadoras completan sus búsquedas, se verifica si hay alguna fuente que no ha sido mejorada a través de un número predeterminado de ciclos, para abandonarla. Las abejas exploradoras eligen una fuente nueva para substituirla.

- Fase de abejas exploradoras (scout)

  Las abejas exploradoras ingresan para descubrir accidentalmente fuentes de alimentos desconocidas. Esta operación puede definirse de la siguiente manera: (4)

$$x_i = x_{min} + rand(0,1).(x_{max} - x_{min})$$

  Este proceso ayuda a evitar soluciones sub-óptimas.

Artificial Bee Colony (ABC)

(1) Generate the initial population x_i (i=1,2,...,SN), using (1)
(2) Evaluate the fitness(fit(x_i)) of the population
(3) g = 0
(4) Repeat
(5) **For** each **employed bee{**
Produce new solutions $v_{i,g}$ in the neighbourhood of x_i, using (2)
Calculate its fitness value fit($v_{i,g}$)
Apply the selection process between x_i and $v_{i,g}$
For each solution (x_i)**{** Calculate the probability values p_i, using (3) **}**
(6) **For** each **onlooker bee{**
Select a solution x_i depending on p_i
Produce new solutions $v_{i,g}$ from the solutions x_i, using (2)
Calculate its fitness value fit($v_{i,g}$)
Apply the selection process**}**
(7) **If** there is an abandoned solution x_i **then**
replace it with a new randomly produced solution $x_i$by the **scout bee**, using (4)
Memorize the best solution achived so far
(8) g = g +1
(9) Until g = max_iter

Artificial Bee Colony (ABC)

Parámetros:

Los parámetros del algoritmo son los siguientes:

1. SN : es el número de fuentes de alimento
2. max_iter: el número total de iteraciones que ejecutará ABC
3. límit: número de ciclos que será conservada una solución (fuente de alimento) puede permanecer sin mejorar antes de ser reemplazada por una nueva solución generada por una abeja exploradora.

Artificial Bee Colony (ABC)

Consideraciones:

- Abejas empleadas: su número es proporcional al número de fuentes de alimento, y su función es evaluar y modificar las soluciones actuales para mejorarlas. Si la nueva posición no es mejor entonces se mantiene la posición actual.

  # Abejas empleadas = # fuentes de alimento

- Abejas en observadoras: su número es proporcional al número de fuentes de alimento. Estas abejas escogerán una fuente de alimento, con base en la información que comparten las abejas empleadas mediante la danza. Esta danza se puede simular con método de la ruleta o torneo de tamaño "t", donde la fuente de alimento con mejor valor de la función objeto es seleccionada.

  # Abejas observadoras = # Abejas empleadas = # fuentes de alimento

Artificial Bee Colony (ABC)

Consideraciones:

- ABC es un algoritmo de naturaleza estocástica que combina la búsqueda local (realizada por las abejas empleadas y observadoras), y también la búsqueda global (por las abejas exploradoras) para equilibrar el proceso de exploración y explotación.
- Las principales ventajas del algoritmo ABC sobre otros métodos de optimización son:
  - Es simple de implementar
  - Tiene pocos parámetros de control
  - Es robusto y altamente flexible
  - Fácil de combinar con otros métodos
  - Rápida convergencia al combinar procesos de exploración y explotación.

Artificial Bee Colony (ABC)

Consideraciones:

- ABC tiene algunas debilidades cuando se pone en práctica.
  - Este método requiere evaluar el fitness en cada nueva fase de abejas del algoritmo para mejorar su rendimiento. Necesita una gran cantidad de evaluaciones de la función objetivo.
  - Se ralentiza cuando la población de soluciones aumenta el costo computacional.
  - Tiene muchas iteraciones y, por lo tanto, requiere una gran capacidad de memoria.

Bibliografía

- E. Bonabeau, et. al , 1999. Swarm Intelligence. From Nature to Artificial Systems. Oxford University Press.
- Eberhart, R. C. and Kennedy, J. A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan. pp. 39-43,1995
- Kennedy, J. and Eberhart, R. C. Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ. pp. 1942-1948, 1995.
- A. Moraglio, C. Di Chio, and R. Poli. Geometric Particle Swarm Optimization. In 10th European conference on Genetic Programming (EuroGP2007), volume 4445 of Lecture Notes in Computer Science. Springer, Abril 2007.

Bibliografía

- D. Karaboga. An idea based on honey bee swarm for numerical optimization. Tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- Raúl Benítez, Gerard Escudero, Samir Kanaan. Inteligencia artificial avanzada.
- M. Dorigo, Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italy, 1992.
- Marco Dorigo, Mauro Birattari, and Thomas Stutzle, Universite Libre de Bruxelles, BELGIUM.
- R. Storn,1997. Differential Evolution, A simple and efficiente heuristic of strategy for global optimization over continuous spaces. Journal of Global Optimization, 11 (1997) 341-359.

¡Gracias!