

Diplomado de especialización de desarrollo de aplicaciones con Inteligencia Artificial

Optimización industrial con Computación Evolutiva

Sesión 5 Inteligencia Colectiva

Dr. Soledad Espezua. LI.
sespezua@pucp.edu.pe

Dr. Edwin Villanueva T.
evillatal@pucp.edu.pe



Agenda

- ▶ Inteligencia Colectiva (IC)
 - ▶ Introducción
 - ▶ Definición de IC
 - ▶ Principales Algoritmos
 - ▶ PSO
 - ▶ ABC
 - ▶ ACO (siguiente sesión)
 - ▶ Otros algoritmos de IC
 - ▶ Bibliografía

Introducción

A look back, a glance ahead

“Nature is concerned with that which works. Nature propagates that which survives. She has little time for erudite contemplation, and we have joined her in her expedient pursuit of betterment”¹

David E. Goldberg



1. Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Reading: Addison-Wesley. Cap. 8, p 309.

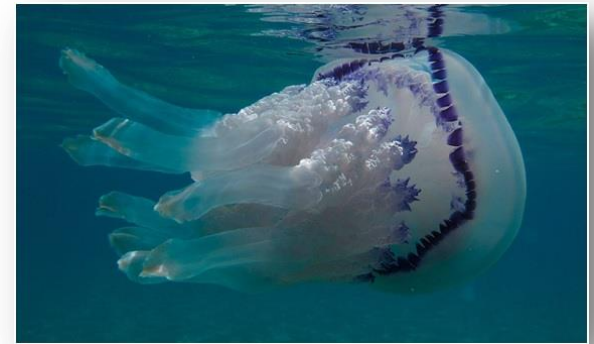
Introducción

- La naturaleza no desarrolla formas de vida inteligente por premeditación.
- Todos los seres vivos desde los más simples a los más complejos, reaccionan ante su entorno.

Pepinos de mar



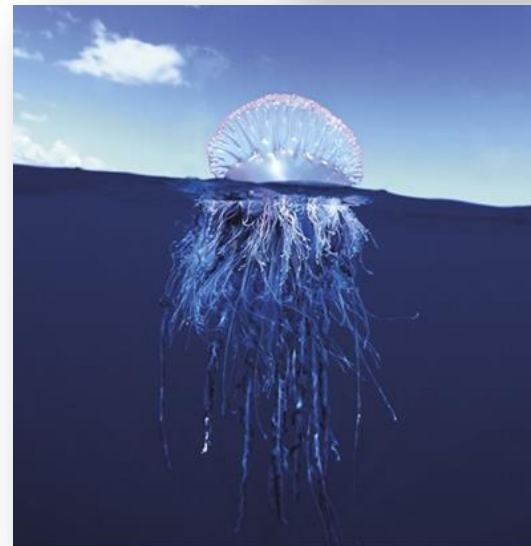
Medusas

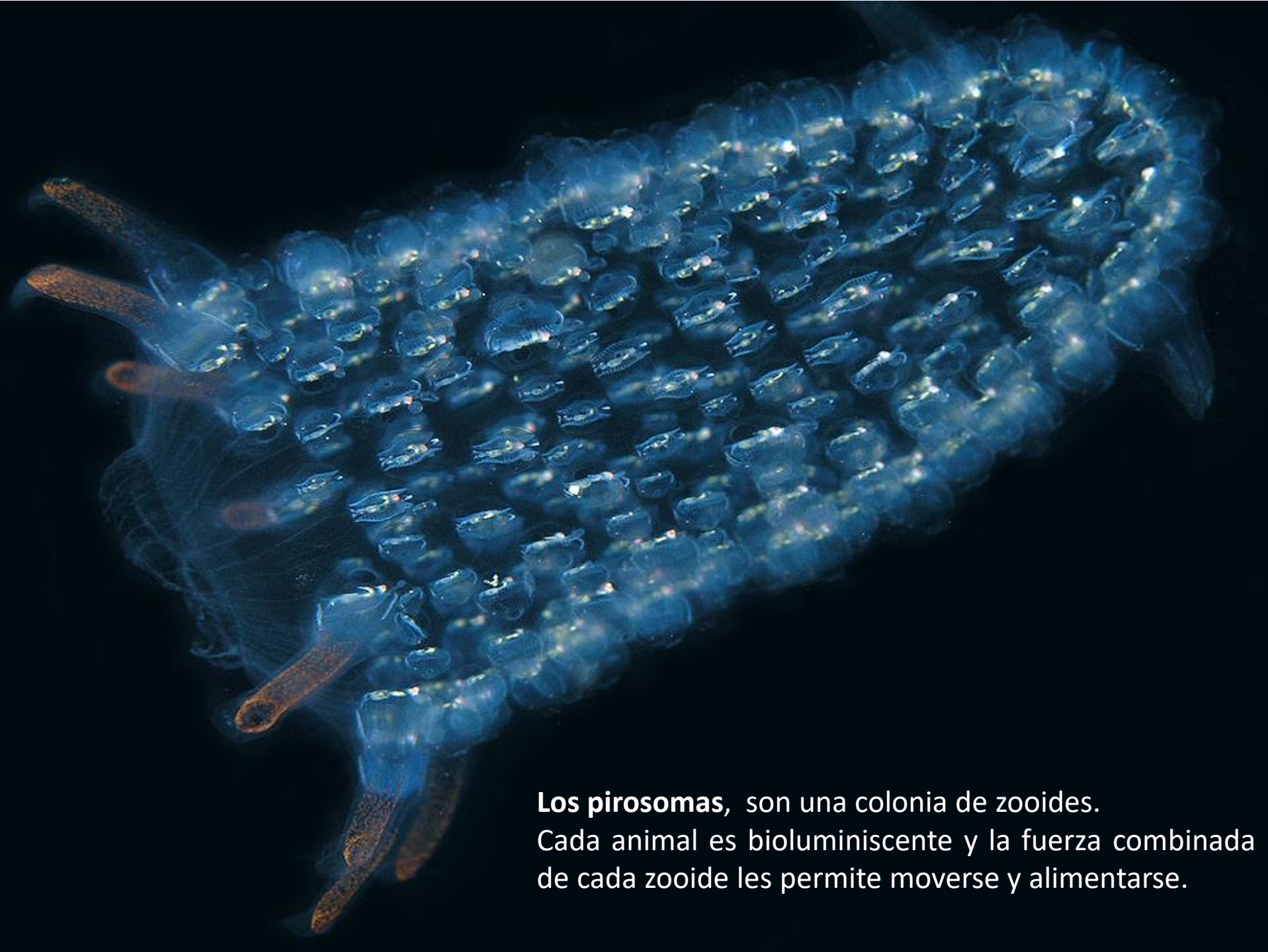


Carabelas portuguesas

Son una colonia de zooides que al nacer se pegan unos a otros y siguen juntos de por vida.

Cada zooide se especializa en una función de supervivencia: defensa, alimentación o reproducción.



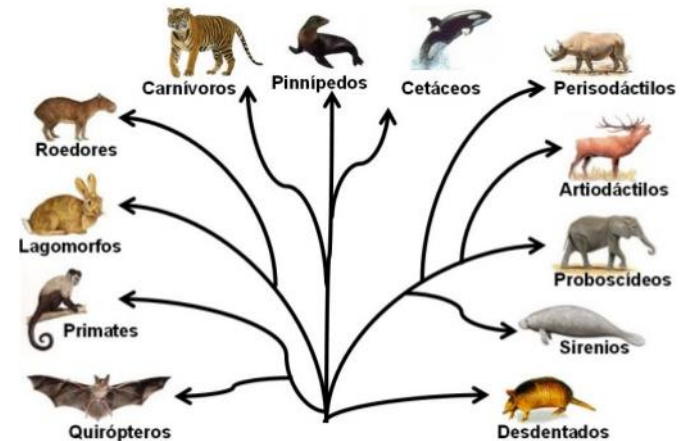


Los pterosomas, son una colonia de zooides. Cada animal es bioluminiscente y la fuerza combinada de cada zooides les permite moverse y alimentarse.

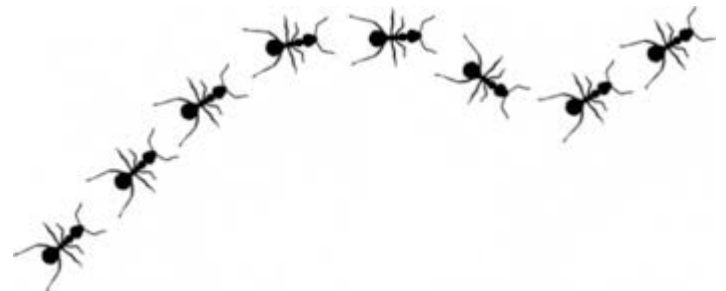
Introducción

Inteligencia

- Filogenética: a nivel de especies (genoma de las especies).



- Ontogenética: a nivel de individuo (memoria neuronal).



- Socio genética: a nivel de grupo (experiencia compartida).



Inteligencia colectiva (*Swarm Intelligence*)

manadas de mamíferos



La inteligencia colectiva (IC) es una rama de la IA que analiza el comportamiento colectivo de sociedades de agentes auto-organizados.

Los algoritmos de IC son meta heurísticas inspiradas en conductas colaborativas, generalmente de la misma especie, observadas en la naturaleza.



**bandadas de aves o
cardúmenes de peces**



**abejas, avispas, hormigas,
termitas, arañas**

sociedades de insectos:



Inteligencia Colectiva (IC)

- ▶ EL termino IC¹, fue introducida por Gerardo Beni 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 hormigas.

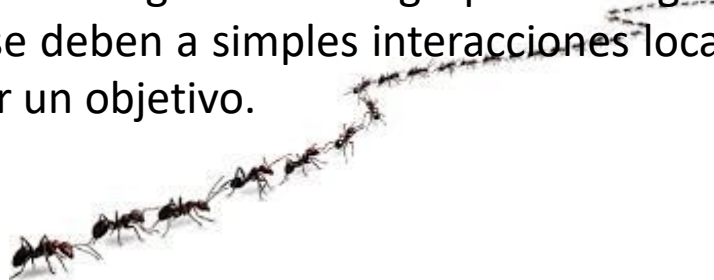
Gerardo Beni



Jing Wang



- ❖ La inteligencia colectiva emerge de un grupo de agentes simples, cuyas capacidades y fortalezas se deben a simples interacciones locales entre los agentes y su entorno para alcanzar un objetivo.



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

IC - Artificial life

Otro ejemplo de IC en la naturaleza se puede ver en los bancos de peces o bandadas de pájaros.

Craig Reynolds, un experto en vida artificial y computación gráfica, observó que el movimiento de bandadas de pájaros se sincronizaba sin que existiese un control central.

Films and Video using Behavioral Animation

1987: Stanley and Stella in: *Breaking the Ice*

1988: *Behave*

1989: *The Little Death*,

1992: [Batman Returns](#)

1993: [Cliffhanger](#)

1994: [The Lion King](#)

1996: [From Dusk Till Dawn](#)

1996: [The Hunchback of Notre Dame](#)



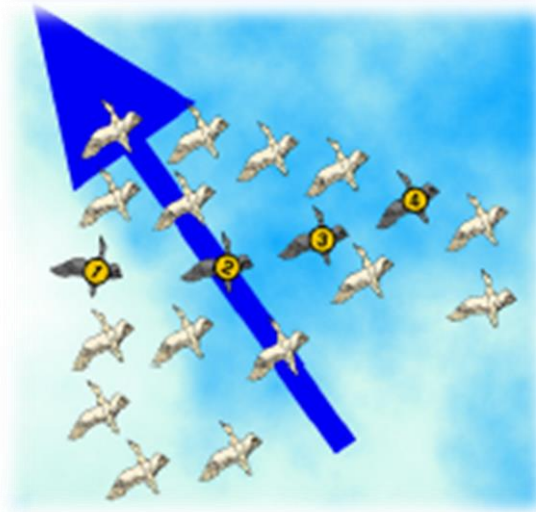
Craig Reynolds

Homepage: <http://www.red3d.com/cwr/>

IC - *Artificial life*

C. Reynolds, observo que las aves seguían 3 reglas:

- 1) Evitar colisiones con sus vecinos
- 2) Tratar de volar a la misma velocidad que sus vecinos
- 3) Tratar de mantenerse cerca de sus vecinos



IC - Artificial life

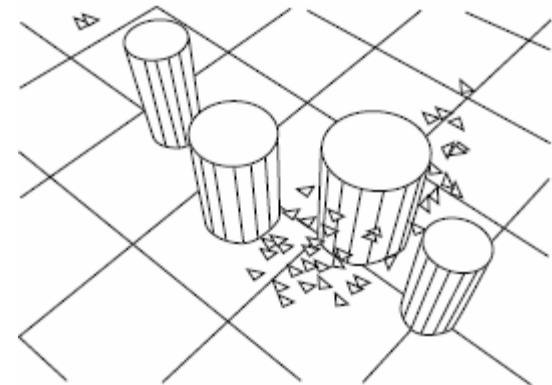
C. Reynolds propuso un **modelo computacional de movimiento de "Boids"** ²:

- 1) **Separación**: evita que los individuos colisionen entre sí
- 2) **Alineación**: dirige a los individuos hacia el mismo rumbo de sus vecinos
- 3) **Cohesión**: evita que los individuos se separen del grupo
- 4) **Desviación**: hace que cada individuo desvíe obstáculos.

Los *boids*¹ de C. Reynold se dividen automáticamente en dos grupos cuando se encuentran con un obstáculo, y se reúnen en un grupo cohesionado después de pasar por él.



Craig Reynolds - Original
(1986) Boids simulation



Otros links: <https://processing.org/examples/flocking.html>

1. Reynolds, Craig W. (1987). "Flocks, herds, and schools: A distributed behavioral model". Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'87). ACM. **21** (4): 25–34.

IC – Características

Los modelos basados en IC:

- ❑ **Crean agentes simples** que siguen reglas simples de comunicación local para gobernar sus acciones y vía un control distribuido realizan interacciones en todo el grupo.
- ❑ No hay un único individuo que controle el grupo.
- ❑ No existe un plan global que rige el comportamiento de la comunidad.
- ❑ La inteligencia de cada individuo hace que tome la mejor decisión utilizando información local. Dichas decisiones también afectan a otros individuos vecinos.
- ❑ **Son robustos**, las acciones siempre se completan a pesar de que un individuo falle.
- ❑ **Son flexibles**, ya que la comunidad puede responder a cambios externos, mediante la percepción del entorno.

IC – Características

Los modelos basados en IC:

- ▶ Buscan resolver problemas de optimización a través de la maximización o minimización de un objetivo (“función objetivo”) el cual puede ser modelado desde una simple ecuación, hasta un complejo sistema de reglas dependiendo del dominio de aplicación.
 - En estos problemas el espacio de búsqueda es muy grande y además, pueden existir varias soluciones.
 - Son problemas conocidos como NP-Complejos.

IC - Principales Algoritmos

- ❖ *Particle Swarm Optimization (PSO)*
- ❖ *Artificial Bee Colony (ABC)*
- ❖ *Ant Colony Optimization (ACO)*

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.

1. *Particle Swarm Optimization (PSO)*

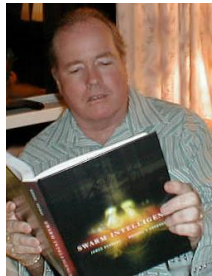
Cúmulos /enjambres/nubes de partículas



Conjunto de métodos inspirados en el comportamiento social de bandadas de aves , cardumen de peces y otros.

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 inspiró en el comportamiento grupal de aves/peces en la naturaleza.
- ▶ Para crear PSO usaron las reglas de cohesión y alineamiento del modelo de C. 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 grupo nunca mueren.

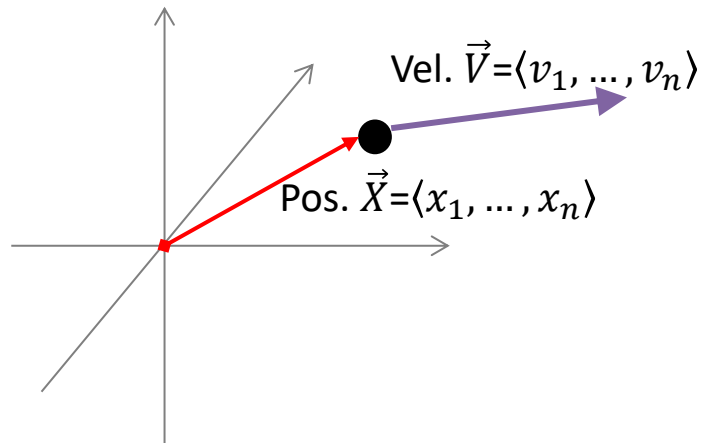
[2] Kennedy, J.; Eberhart, R. (1995). Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks Vol. IV: 1942–1948. .

PSO - Características

- ▶ **Una población esta formada por un conjunto de partículas** que representan las soluciones a un problema.
- ▶ Las partículas son simples agentes que vuelan a través del espacio de búsqueda y almacenan o comunican, la mejor solución que han descubierto.
- ▶ Todas las partículas se mueven por el espacio del problema con el objetivo de encontrar una solución óptima global.
- ▶ Cada partícula se representa con un vector de posición llamado \vec{X} . La longitud de este vector es igual al número de variables del problema.
- ▶ Las variaciones de posición de las partículas, se consiguen utilizando otro vector llamado vector de velocidad (\vec{V}), que tiene la misma longitud que \vec{X} .

PSO - Características

- ▶ Una partícula en PSO tiene:
 - Un vector \vec{X} de su posición actual.
 - Un vector \vec{V} de velocidad que dirige su movimiento.
 - La capacidad de intercambiar información con sus vecinos.
 - La capacidad de memorizar una posición anterior.



PSO - Movimiento de partículas

- La ecuación para **actualizar la posición** propuesta para este algoritmo es:

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

Una partícula i se mueve a una nueva posición x_i^{k+1} , dependiendo de su posición actual (x_i^k) y su nueva velocidad (v_i^{k+1}).

- La ecuación para **actualizar la velocidad** es la siguiente:

$$v_i^{k+1} = \underbrace{\omega \cdot v_i^k}_{\text{Fuerza inercial}} + \underbrace{\varphi_1 \cdot \text{rnd}_1 \cdot (pBest_i - x_i^k)}_{\text{Fuerza hacia la mejor posición local } pBest_i} + \underbrace{\varphi_2 \cdot \text{rnd}_2 \cdot (g - x_i^k)}_{\text{Fuerza hacia la mejor posición global } g \text{ del enjambre.}} \quad (2)$$

La velocidad con la que cada partícula se mueve se ajusta basándose en la mejor posición que ha obtenido ella misma (aprendizaje cognitivo) y la mejor solución global (aprendizaje social) encontrada hasta el momento por el grupo.

PSO - Movimiento de partículas

Nueva Posición: $x_i^{k+1} = x_i^k + v_i^{k+1}$

Inercia

Memoria

Cooperación

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

donde:

- k , iteración
- i , partícula
- v_i^k , velocidad
- ω , inercia de la partícula
- $pBest_i$, mejor posición encontrada por partícula i
- g , mejor posición **global** encontrada por el enjambre (*swarm*)
- φ_1 y φ_2 , pesos de tendencia individual o social respectivamente.
- rnd_1 y rnd_2 , valores aleatorios entre $\{0,1\}$.

PSO - Elección de parámetros ($\omega, \varphi_1, \varphi_2$)

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

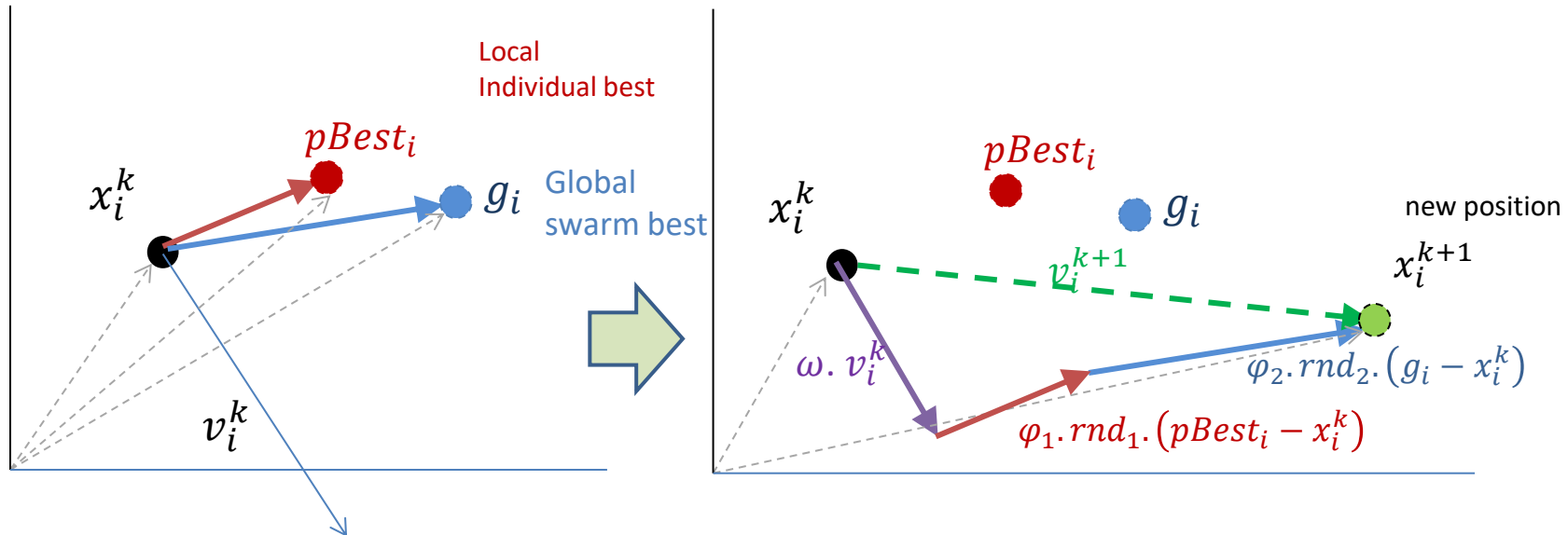
- ▶ Algunas recomendaciones de valores para:
 - Inercia de la partícula $\omega = 0.5$, porque debe influir menos.
 - Para $(\varphi_1, \varphi_2) = 2$ (tendencia local φ_1 y tendencia global del enjambre φ_2).
- ▶ Una mejora del método consiste en empezar con una inercia relativamente alta ($\omega = 1.4$) que se va reduciendo en cada iteración, por ejemplo, multiplicándola por un factor $r < 1$ para estabilizar el sistema.
- ▶ Otras propuestas:

$$\text{inercia } \omega = (\omega_1 - \omega_2) \cdot \left(\frac{\text{maxiter} - \text{iter}}{\text{iter}} \right) + \omega_2$$

$$\text{inercia } \omega_i = \left(1.1 - \frac{gbest_i}{pbest_i} \right)$$

} Hacer decaer linealmente la inercia en el tiempo

PSO - Movimiento de una partícula



$$v_i^{k+1} = \underbrace{\omega \cdot v_i^k}_{\text{Inercia}} + \underbrace{\phi_1 \cdot rand_1 \cdot (pBest_i - x_i^k)}_{\text{Memoria}} + \underbrace{\phi_2 \cdot rand_2 \cdot (g_i - x_i^k)}_{\text{Cooperaci3n}} \quad (2)$$

Aprendizaje Cognitivo Aprendizaje Social

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

Pseudocódigo PSO

Algorithm Pseudocodigo del algoritmo PSO Canónico

```
1:  $S \leftarrow \text{SwarmInitialization}()$ 
2: while not stop condition do
3:   for each particle  $x_i$  of the swarm  $S$  do
4:     evaluate( $x_i$ )
5:     if  $\text{fitness}(x_i)$  is better than  $\text{fitness}(pBest_i)$  then
6:        $pBest_i \leftarrow x_i$ 
7:     end if
8:     if  $\text{fitness}(pBest_i)$  is better than  $\text{fitness}(g_i)$  then
9:        $g_i \leftarrow pBest_i$ 
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
```

Consideraciones del Procedimiento

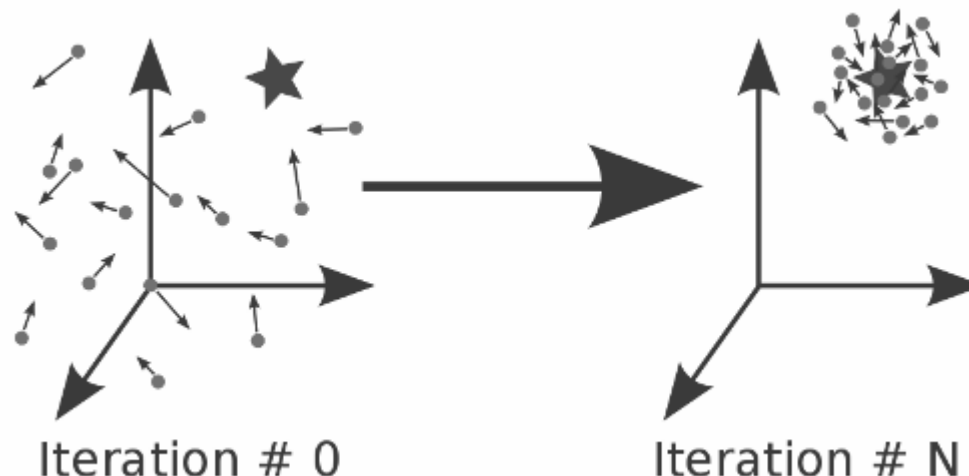
- Población inicial: crea un conjunto S de partículas, donde a cada partícula x_i se le asigna una posición inicial y una velocidad, ambas deben estar acotadas por límites:

$$x_i \leftarrow \cup (inf, sup)$$

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

- Iteraciones:

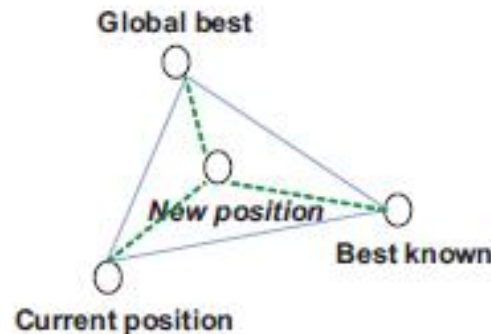
- Después de actualizar todas las partículas, se comparan sus mejores posiciones $pBest$ con la mejor posición global g , si algún $pBest$ es mejor que g , entonces: $g \leftarrow pBest_i$.
- Al acabar las iteraciones g contiene la mejor solución encontrada por el algoritmo



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 formuló para trabajar con representación real, hay 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 geométrico. El algoritmo puede trabajar con soluciones binarias utilizando la distancia de *Hamming*. También puede trabajar con soluciones de representación entera, utilizando la distancia *Manhattan* y para soluciones con representación continua se utilizara la distancia *euclidiana*.

Representación de GPSO³,
(izquierda) partículas en el espacio de
influencia geométrico y
(derecha) partículas
sobre el espacio de búsqueda



[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.

PSO



2. *Artificial Bee Colony (ABC)*



Conjunto de métodos inspirados en las actividades de una colonia de abejas.



Artificial Bee Colony (ABC)

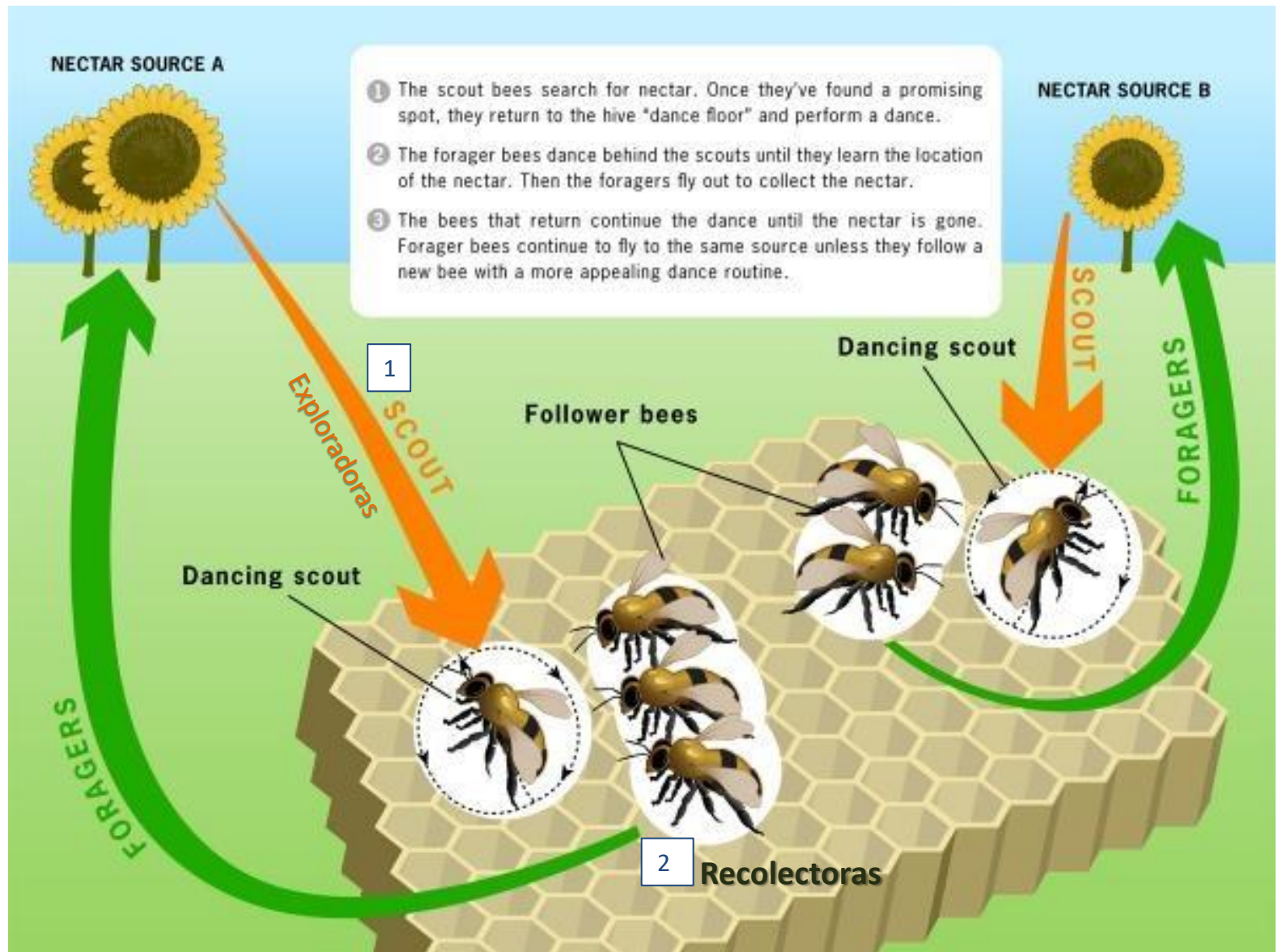
- ▶ El algoritmo de colonias de abejas o *Artificial Bee Colony* (ABC) ⁶, fue presentado por **Dervis Karaboga** en 2005.



- ▶ ABC es inspirado en el comportamiento observado en las abejas domésticas.
- ▶ ABC es un algoritmo propuesto para resolver problemas de optimización combinatoria. Está basado **en poblaciones** en la cual **los individuos, son las posiciones de comida**, y se modifican por acción de las abejas artificiales.
 - El objetivo de las abejas es descubrir los lugares de comida con néctar, hasta encontrar un lugar con la 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.

Modelo Biológico



ABC - Modelo

- ▶ En un modelo ABC, las abejas se mueven por el espacio de búsqueda eligiendo fuentes de néctar (individuos de la población) dependiendo de su experiencia pasada y de su compañera de colmena.
- ▶ Cada fuente de néctar/ alimento es una solución alternativa al problema de optimización, y la cantidad de néctar de una fuente es el *fitness* de la solución.



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.

ABC y el Modelo biológico

- ▶ El modelo biológico de abejas guía a sus individuos hacia la búsqueda de néctar y el modelo artificial ABC es un proceso de **optimización**. El modelo ABC consta de los siguientes elementos:

- 1) **Fuente de néctar/alimento:** Es un vector (posición) de valores reales.
- 2) **Abejas recolectoras empleadas (Forager):** Traen el alimento a la colmena. Comunican su ubicación y rentabilidad, con una cierta probabilidad, a sus compañeras observadoras.
- 3) **Abejas desempleadas:** Están en constante búsqueda de fuentes de alimento. Hay dos tipos:

Exploradoras (Scout): se encargan de buscar en el entorno que rodea a la colmena nuevas fuentes de alimento. Comunican su ubicación mediante una danza (*fitness*).

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



ABC - Algoritmo básico

- ▶ **Población inicial:** es iniciada con un número SN (*Source Number*) de fuentes de alimento, donde cada fuente es un vector 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 (*employee bee*):**

En esta fase la búsqueda se realiza cerca de cada solución, una por una. Cuando las abejas empleadas visitan fuentes de alimento, calculan una nueva posición $v_{i,g}$, por eso estas abejas representan operadores de variación:

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

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 , iteración actual
- r , número real aleatorio entre $[-1, 1]$.

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 $fitness(v_{i,g}) > fitness(x_i)$
 $x_i \leftarrow v_{i,g}$ se convertirá en un nuevo miembro de la población; de lo contrario x_i se conserva.

ABC - Algoritmo básico

► Fase de selección por las abejas observadoras (*onlooker bee*)

En esta fase, 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 al valor de su *fitness* (calidad de néctar que tiene la solución). Esta es la propiedad de retroalimentación positiva del algoritmo ABC. Se calcula de la siguiente manera:

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

donde:

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

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

- Luego 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)

ABC - Algoritmo básico

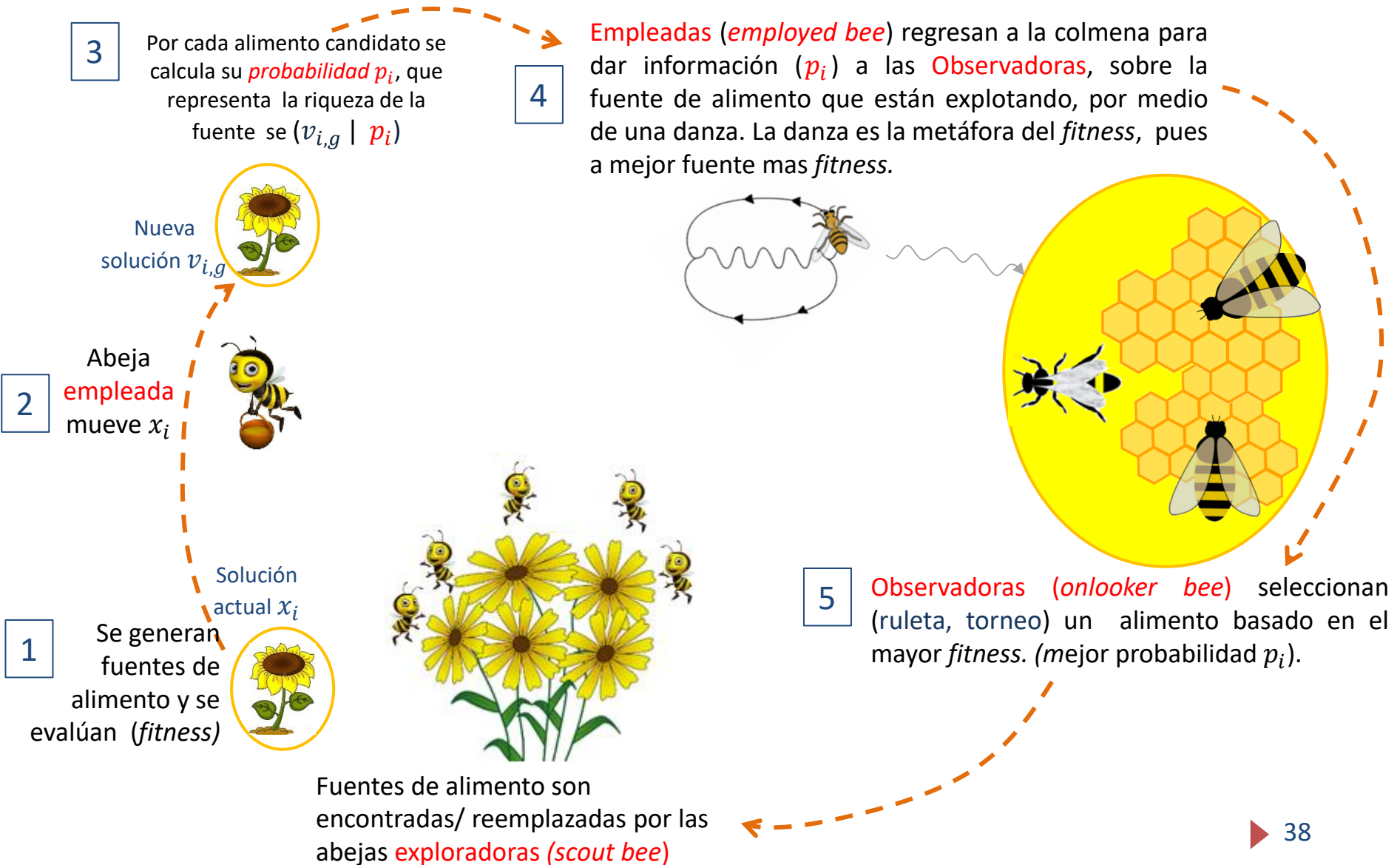
- ▶ Cuando todas las abejas **empleadas** y **observadoras** completan sus búsquedas, se verifica si hay alguna fuente que no ha sido mejorada a través de un número predeterminado de ciclos, para abandonarla. Las **abejas exploradoras eligen una fuente nueva**.
- ▶ **Fase de abejas exploradoras (*scout*):**

Las abejas exploradoras ingresan para generar nuevas fuentes de alimento de manera aleatoria y para substituir fuentes que no han sido mejoradas. Esta operación puede definirse de la siguiente manera:

$$x_i = x_{min} + rand(0,1) \cdot (x_{max} - x_{min}) \quad (4)$$

Este proceso ayuda a evitar soluciones sub-óptimas.

ABC - modelo



ABC - Parámetros del Algoritmo básico

Los parámetros del algoritmo son los siguientes:

1. **SN:** número de fuentes de alimento (tamaño de la población)
2. **max_iter:** número total de iteraciones que ejecutará ABC
3. **Limit:** número de ciclos que será conservada fuente de alimento (una solución sin mejorar) antes de ser reemplazada por una nueva solución generada por una abeja exploradora.

ABC - Pseudocódigo

- (1) Generate the initial population x_i ($i=1,2,\dots,SN$), using (1)
- (2) Evaluate the fitness($\text{fit}(x_i)$) of the population
- (3) $g = 0$
- (4) **While** ($g < \text{max_iter}$)
- (5) **For each employed bee**{
 - Produce new solutions $v_{i,g}$ in the neighbourhood of x_i , using (2)
 - Calculate its fitness value $\text{fit}(v_{i,g})$
 - Apply the selection process between x_i and $v_{i,g}$
- (6) **For each solution** (x_i) { Calculate the probability values p_i , using (3) }
- (7) **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 $\text{fit}(v_{i,g})$
 - Apply the selection process
- (8) **If Limit==true**
 - there is an abandoned solution x_i replace it with a new randomly produced solution x_i for the **scout bee**, using (4)
- (9) Memorize the best solution achieved so far
- (10) $g = g + 1$

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 solución no es mejor entonces mantiene la solución actual.

Abejas empleadas = # fuentes de alimento

- ▶ **Abejas 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 mediante el método de la ruleta o torneo de tamaño "t", donde la fuente de alimento con mejor *fitness* es seleccionada.

Abejas observadoras = # Abejas empleadas = # fuentes de alimento

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**.
 - Empleadas y Observadoras (operadores de variación)
 - Exploradoras (operadores de reemplazo)
- ▶ 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.

ABC - Consideraciones

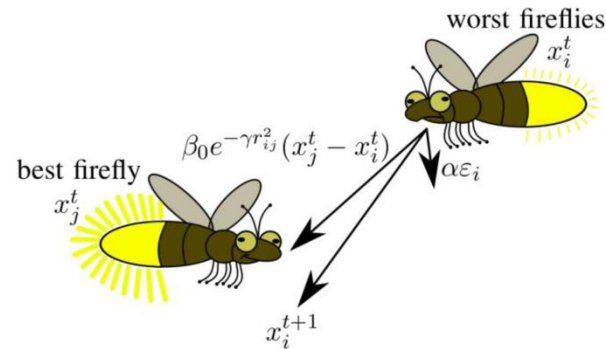
- ▶ ABC tiene algunas debilidades cuando se pone en práctica.
 - Este método requiere evaluar el *fitness* en cada nueva fase de abejas del algoritmo para mejorar su rendimiento. Necesita una gran cantidad de evaluaciones de la función objetivo.
 - Se ralentiza cuando la población de soluciones aumenta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

ABC



IC - Otros Algoritmos

○ Firefly Algorithm (FA)



Es un algoritmo inspirado en las **luciérnagas** y sigue un modelo simplificado, donde:

- La población son luciérnagas que brillan (*fitness*) con cierta intensidad
- Todas las luciérnagas se atraen proporcionalmente a su brillo, la menos brillante se moverá hacia la más brillante (atractivo).
- El atractivo y el brillo disminuyen exponencialmente a medida que aumenta su distancia y un coeficiente de absorción de luz definido. Si no hay ninguna luciérnaga brillante, todas se mueven al azar.
- Nuevas soluciones son evaluadas y actualizadas su intensidad. La solución es obtenida de un ranking de *fitness* al final de un ciclo poblacional.

IC - Otros Algoritmos

○ *Spider Monkey Optimization (SMO),*



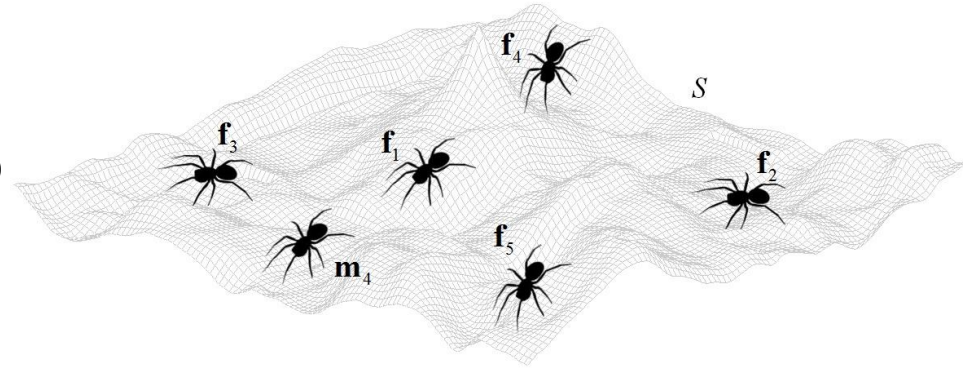
Para buscar alimento, **los monos araña** trabajan en un grupos grandes y según la necesidad, después de un tiempo, se dividen en grupos más pequeños dirigidos por una hembra adulta.

- SMO es un algoritmo basado en la conducta social de los monos araña para buscar alimentos (función objetivo).
- Una población, dirigida por la mejor hembra (experiencia global), se fragmenta en pequeños grupos (experiencia local) para buscar comida.
 - En caso de que la hembra no cumpla con el objetivo, el grupo se subdivide, una vez más, con una hembra a la cabeza y replicando el proceso hasta llegar a la comida.
- Para actualizar sus posiciones, se llevan a cabo varios pasos: inspección amplia del espacio de búsqueda y selección de resultados.

IC - Otros Algoritmos

○ *Social Spider Optimization (SSO)*

El algoritmo SSO se basa en la simulación del comportamiento cooperativo de las **arañas sociales**.



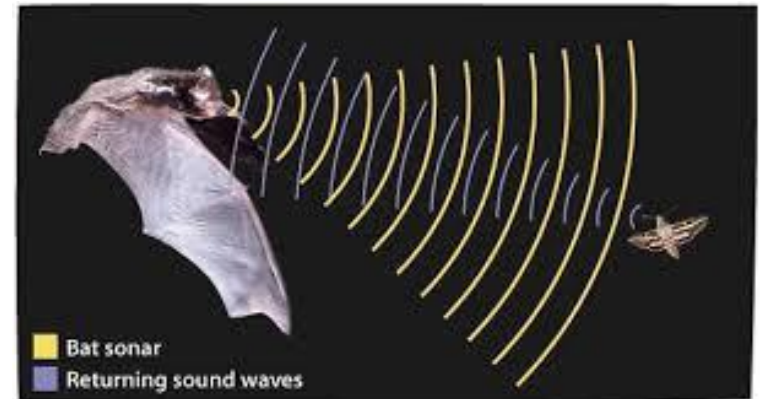
- Los individuos emulan un grupo de arañas que interactúan entre sí según las leyes cooperativas de la colonia.
- El algoritmo considera dos agentes de búsqueda diferentes (arañas): machos y hembras.
 - Dependiendo del género, cada individuo es conducido por un conjunto de operadores evolutivos diferentes que imitan comportamientos cooperativos que típicamente se encuentran en la colonia.

Sakshi A. et al. "A Social Spider Optimization Algorithm with Chaotic Initialization for Robust Clustering". *Procedia Computer Science*. Volume 143, 2018, Pages 450-457

IC - Otros Algoritmos

○ *Swarm Bat Algorithm (ABA)*

Este algoritmo está inspirado en el comportamiento de caza de los **murciélagos**. Los cuales usan eco-localización a diferentes frecuencias y emisiones de volumen.



- Una población se compone de murciélagos que vuelan al azar, tienen velocidad v_i , posición x_i y frecuencia fija f_{min} .
- Se generan nuevas soluciones ajustando la frecuencia, actualizando la velocidad y la posición.
- Si una determinada frecuencia es superada entonces se escoge la mejor solución; y se genera una nueva solución alrededor de la seleccionada. En cada solución se varían la longitud de onda λ de la frecuencia y el volumen para buscar presas (función objetivo).

Wang, X., W. Wang and Y. Wang, 2013b. An Adaptive Bat Algorithm. In Intelligent Computing Theories and Technology.

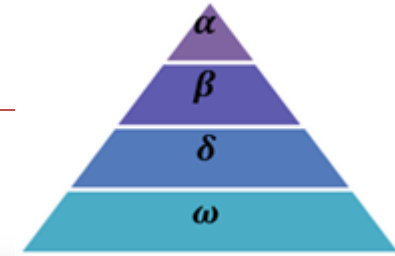
IC - Otros Algoritmos

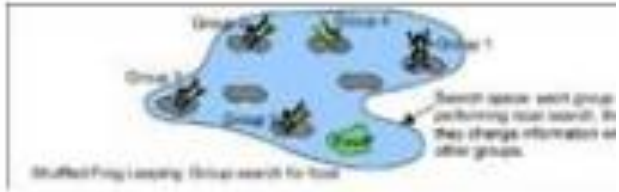
○ Grey Wolf Optimizer (GWO)

Es un algoritmo que se basa en la jerarquía y liderazgo del **lobo gris** para cazar presas.

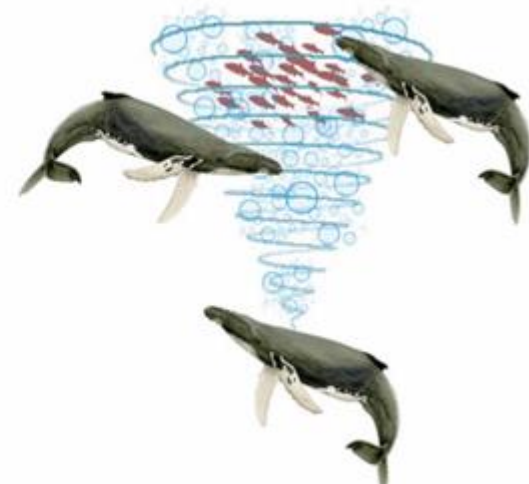
- GWO es un algoritmo simple, poblacional que simula el liderazgo de una manada.
- Después de evaluar su *fitness*, una población se compone de cuatro clases de lobo: alfa, beta, delta y omega,
- La estrategia de caza (función objetivo) de los lobos grises se puede modelar matemáticamente aproximando la posición de la presa con la ayuda de lobos α ; β y δ (soluciones). Cada lobo en la población puede actualizar sus posiciones.

Existen versiones multiobjetivo (Multi-objective Grey Wolf Optimizer: MOGW) bastante usadas en la industria y la ciencia.





-
- The first system of the musical score for 'The Rose Tree' consists of two staves. The upper staff is in treble clef and the lower staff is in bass clef. The key signature has one flat (B-flat), and the time signature is common time (C). The melody in the upper staff begins with a quarter note G4, followed by a quarter note A4, a quarter note B-flat4, and a quarter note G4. The lower staff provides a simple harmonic accompaniment with half notes.

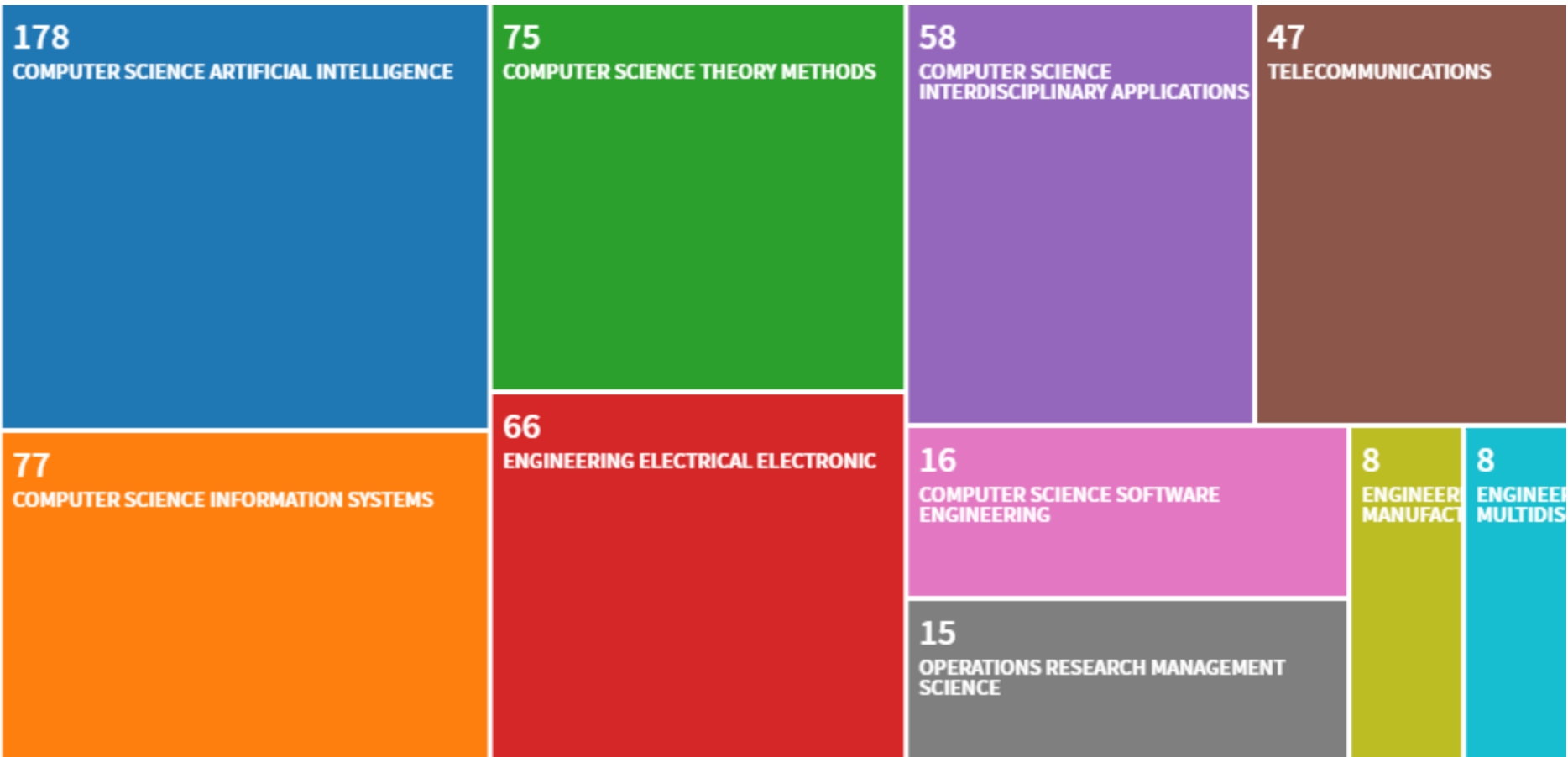


IC - Aplicaciones

Author	Title - Year (2019)
Boveiri, H. R.	An efficient <u>Swarm-Intelligence</u> approach for task scheduling in cloud-based internet of things applications Journal of Ambient Intelligence and Humanized Computing
Choong, S. S.	An <u>artificial bee colony algorithm</u> with a Modified Choice Function for the traveling salesman problem Swarm and Evolutionary Computation
Cruz, D. P. F.	A critical discussion into the core of swarm intelligence algorithms Evolutionary Intelligence
Emary, E.	Feature selection via Levy <u>Ant lion</u> optimization Pattern Analysis and Applications
Ezugwu, A. E.	Symbiotic organisms search algorithm : Theory, recent advances and applications Expert Systems with Applications
Figueiredo, E.	<u>Swarm intelligence</u> for clustering - A systematic review with new perspectives on data mining Engineering Applications of Artificial Intelligence
Gupta, S.	A novel Random Walk Grey Wolf Optimizer Swarm and Evolutionary Computation

Minería de Datos; Enrutamiento de Vehículos con Ventanas de Tiempo; Problema del Agente Viajero; Enrutamiento en redes de comunicación; Entrenamiento de Redes Neuronales; Control de robots.

IC -2019



Bibliografía

- ▶ Ivan Zelinka; Guarong Chen. **Evolutionary Algorithms, Swarm Dynamics and Complex Networks. Methodology Perspectives and Implementation.** Springer-Verlag, 2018
- ▶ Raúl Benítez; Gerard Escudero; Samir Kanaan. **Inteligencia artificial avanzada.** Universitat Oberta de Catalunya (UOC), 2016
- ▶ Silja Meyer-Nieberg; Nadiia Leopold; Tobias Uhlig. **Natural Computing for Simulation-Based Optimization and Beyond.** Springer, 2019
- ▶ Seyedali Mirjalili; Jin Song Dong. **Multi-Objective Optimization using Artificial Intelligence Techniques.** Springer briefs in applied sciences and Technology and computational intelligence, 2019
- ▶ D. Karaboga. An idea based on honey bee swarm for numerical optimization. Tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- ▶ 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

Links interesantes

PSO <http://www.particleswarm.info/>

ABC <https://abc.erciyes.edu.tr/>

Artificial life <http://www.alife.org>

ACO <http://iridia.ulb.ac.be/~mdorigo/ACO/ACO.html>

Differential evolution <http://www1.icsi.berkeley.edu/~storn/code.html>