



# ***Algoritmos Bioinspirados***

## ***Unidad V : Inteligencia de enjambre***

***Programa: Ingeniería en Inteligencia Artificial***

Al ver bandadas de aves o bancos de peces es posible pensar que las acciones de cada individuo están coordinadas con las acciones de los otros.

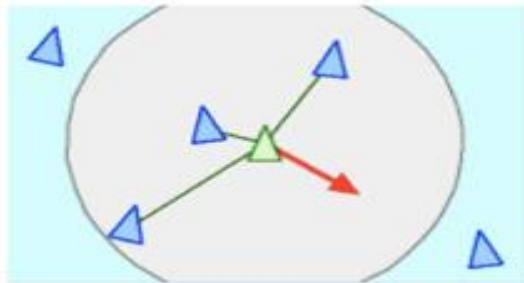
Investigadores afirman que los estorninos coordinan sus movimientos basados en 7 vecinos [5]. Miles de estorninos pueden formar una bandada, pero los movimientos de la bandada se basan en las interacciones locales de grupos superpuestos de solo siete aves.



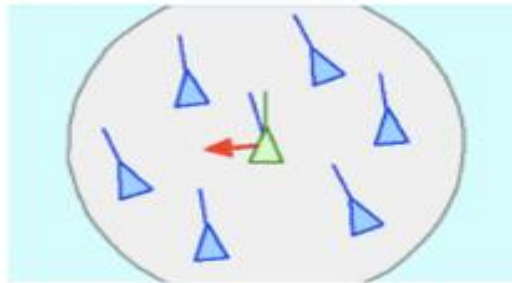
Es interesante que modelar la forma en que las aves se congregan puede conducir a un algoritmo de optimización. Es probable que los organismos que podríamos modelar en estos casos optimicen algo, como la cohesión del grupo versus el esfuerzo individual.

## Antecedentes: *Boids*

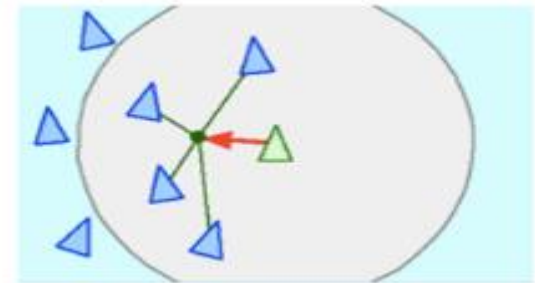
En 1987, Craig Reynolds creó un modelo de movimiento animal coordinado en el que los agentes (boids) obedecían tres reglas locales simples, ahora conocidas como *reglas de Reynolds*.



1. **Separación:** maniobrar para evitar acercarse demasiado a los demás.



2. **Alineación:** maniobrar hacia la dirección promedio de los demás

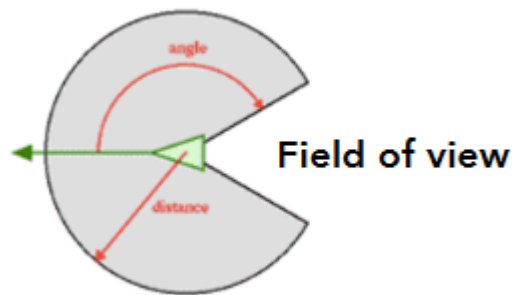


3. **Cohesión:** maniobrar hacia la posición promedio de los demás.

Esta receta logra un comportamiento de enjambre realista, con algunos ajustes de parámetros necesarios dependiendo de la especie simulada

Reynolds, C. (1987). Flocks, herds and schools: A distributed behavioral model. Computer Graphics, vol. 21(4), pp. 25–34.

Es importante notar que cada *boi*d tiene su propio campo de percepción, es decir, que solo puede ver a cierta distancia y con un campo de visión específico (i.e., los *boi*ds no pueden ver detrás de ellos)



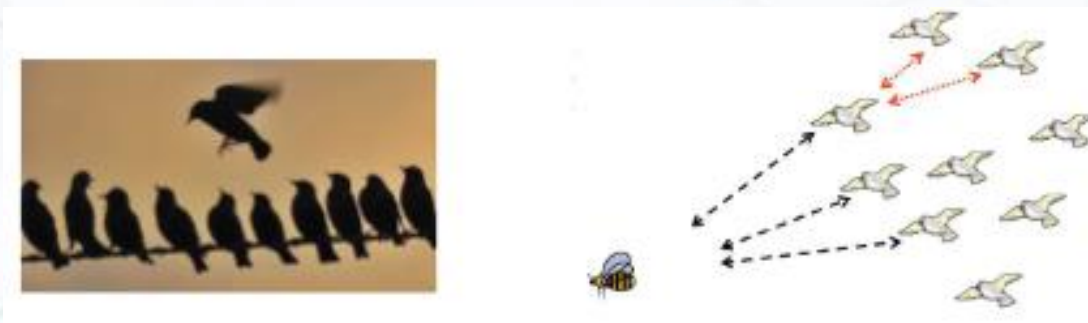
Por tanto, los ajustes que hace a su velocidad en cualquier momento son en función de las posiciones y velocidades de los cuerpos en su campo de percepción, más que una función de toda la bandada.

Es importante tener en cuenta que estas reglas no están estrictamente inspiradas en la naturaleza, ya que Reynolds no intentaba explicar el comportamiento natural del enjambre, simplemente intentaba emularlo. Varios autores señalaron que "muchas personas que ven las simulaciones, las reconocen de inmediato como una representación de una bandada natural y las encuentran igualmente agradables de ver".

<https://www.youtube.com/watch?v=QbUPfMXXQIY>

## *Boids + Roosting behavior*

En 1995, Kennedy y Eberhart incluyeron un dormitorio (*roost*) o, más generalmente, un punto de atracción (ejemplo, una presa) en una simulación similar a *Boids* simplificada.



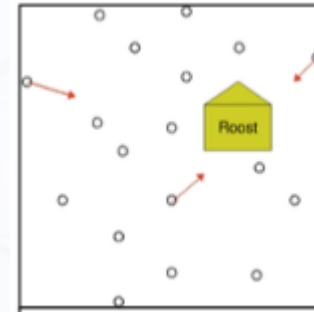
Tal que cada agente:

- Se siente atraído por la ubicación del dormitorio,
- Recuerda cuando estuvo más cerca del dormitorio,
- Comparte información con sus vecinos sobre su ubicación más cercana al dormitorio.

J. Kennedy, R.C. Eberhart. (1995). Particle swarm optimization. In Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp.1942-1948.

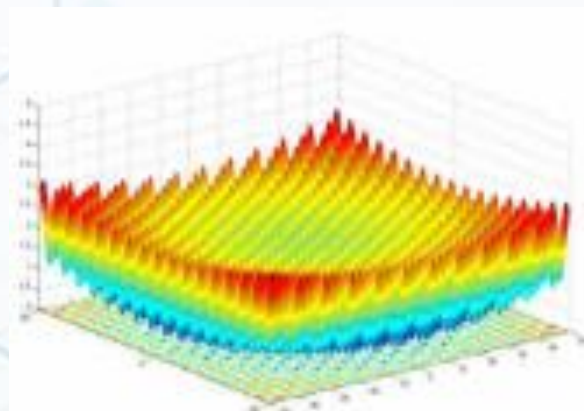


Eventualmente, (casi) todos los agentes aterrizan en el dormidero



Entonces, si:

- dormidero = extremo desconocido (máximo, mínimo) de una función
- distancia al gallinero = calidad de la posición actual del agente

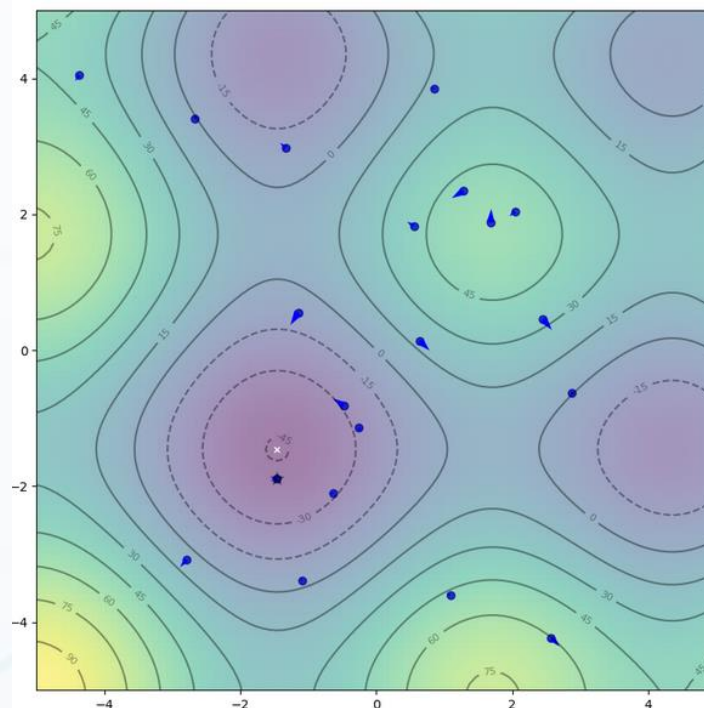


## Optimización de Enjambre de Partículas (PSO)

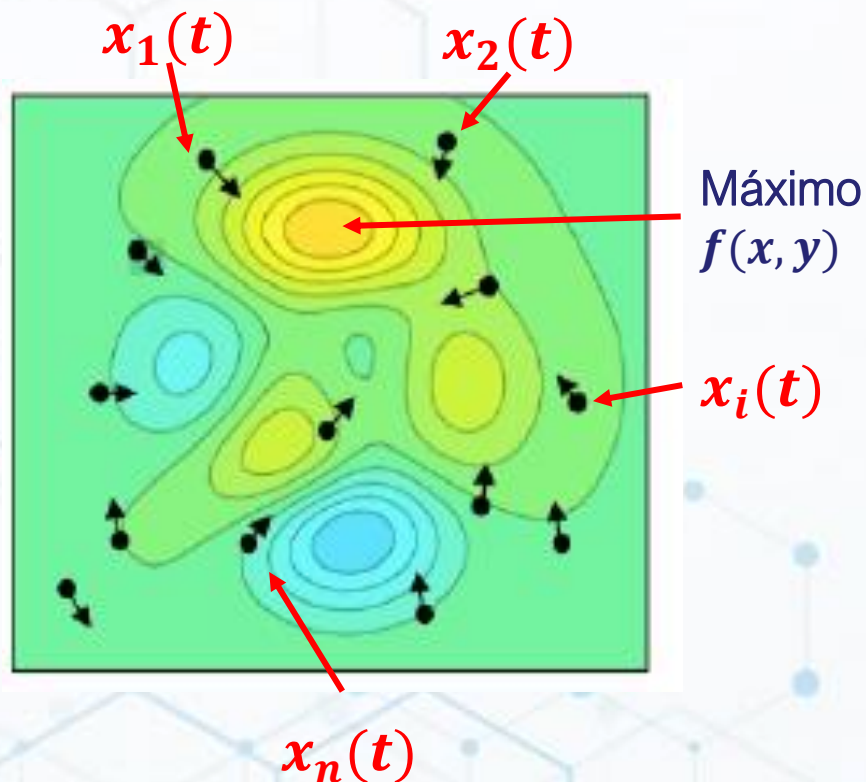
PSO consiste en un sistema multiagente (partículas) que simulan el comportamiento de enjambre de las aves.

- ✓ Una **partícula** se iguala con una solución candidata a un problema de optimización.  $x_i(t)$  denota la partícula  $i$  en el tiempo  $t$ .
- ✓ Tal partícula tiene una posición y una velocidad.
- ✓ Su posición es, de hecho, la solución candidata.
- ✓ Su velocidad es un vector de desplazamiento en el espacio de búsqueda, que (se espera) contribuirá a un cambio fructífero en su posición en la siguiente iteración.

$$f(x, y) = x^2 + y^2 + (25 * (\text{sen}x + \text{sen}y))$$
$$0 \leq x \leq 5, 0 \leq y \leq 5$$



- Cada partícula es denotada por  $x_i(t)$  y representa una solución dentro del espacio de búsqueda.
- El valor de aptitud de cada partícula representa la calidad de su posición en el panorama de optimización.



$$f(x, y) = x^2 + y^2, 0 \leq x \leq 5, 0 \leq y \leq 5$$

$$x_1(t) = (1.3, 2) \rightarrow f(x, y) = 5.69$$

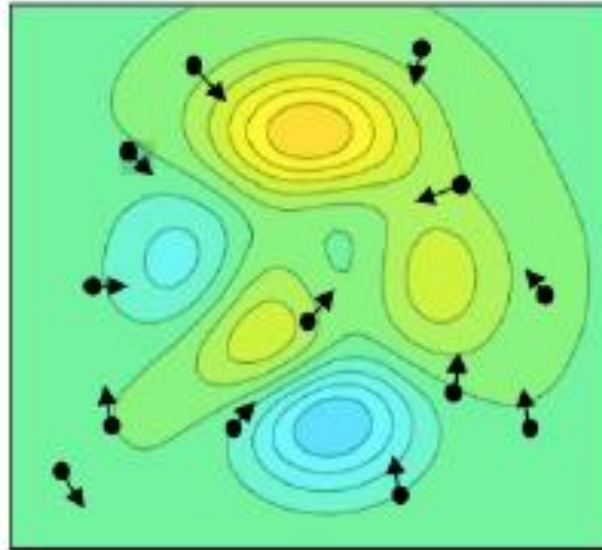
$$x_2(t) = (4, 2.3) \rightarrow f(x, y) = 21.29$$

$$x_3(t) = (3.1, 2.8) \rightarrow f(x, y) = 17.45$$

Para un problema de maximización la partícula  $x_2$  alcanzó el mejor resultado en el tiempo  $t$ .



- Las partículas se mueven sobre el espacio de búsqueda con una cierta velocidad..
- Cada partícula tiene: Estado interno + estado del vecindario (red de conexiones sociales)



$$\{\vec{x}, \vec{v}, \vec{x}_{pbest}, \mathcal{N}(p)\}$$

En cada paso de tiempo, la velocidad (tanto la dirección como la rapidez) de cada partícula se ve influenciada por:

***pbest:*** su propia mejor posición encontrada hasta ahora

***lbest:*** la mejor solución encontrada hasta ahora por los compañeros de equipo en su vecino social, o

***gbest:*** la mejor solución global hasta el momento

más un elemento de aleatoriedad.

$$f(x, y) = x^2 + y^2, 0 \leq x \leq 5, 0 \leq y \leq 5$$

$$\mathcal{N}(x_1) = (x_3, x_5, x_6)$$

$$x_1(t) = (1.3, 2) \rightarrow f(x, y) = 5.69$$

$$x_2(t) = (4, 2.3) \rightarrow f(x, y) = 21.29$$

$$x_3(t) = (1, 5) \rightarrow f(x, y) = 26$$

$$x_4(t) = (3.1, 2.8) \rightarrow f(x, y) = 17.45$$

$$x_5(t) = (4.8, 2.4) \rightarrow f(x, y) = 28.8$$

$$x_6(t) = (2.5, 2.5) \rightarrow f(x, y) = 12.5$$

$$x_7(t) = (5, 2.2) \rightarrow f(x, y) = 29.84$$

$$x_8(t) = (1.9, 3.4) \rightarrow f(x, y) = 15.17$$

$$pbest(x_1) = 5.69$$

$$lbest(x_1) = 28.8$$

$$gbest(x_1) = 29.84$$



### Pasos de PSO:

1. Definir función objetivo.
2. Definir parámetros de ejecución.
  - Número de partículas  $N$
  - Número de pasos de tiempo
  - $a \rightarrow$  inercia
  - $b_1 \rightarrow$  factor de aprendizaje (influencia propia)
  - $b_2 \rightarrow$  factor de aprendizaje (influencia social)
  - Número de vecinos (versión local)
3. Inicialización aleatoria de posiciones y velocidades.
4. Evaluar la función de aptitud con las posiciones actuales de las partículas.
5. Asignar los mejores resultados (pbes, lbest/gbest).



## Pasos de PSO (2):

### 1. Repetir hasta alcanzar condición de paro:

- i. Actualizar la posición de cada partícula de acuerdo con 1 de las siguientes formulas

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

$$v_i(t + 1) = a * v_i(t) + b_1 r_1 (pbest - x_i(t)) + b_2 r_2 (lbest - x_i(t)) \text{ (versión local)}$$

$$v_i(t + 1) = a * v_i(t) + b_1 r_1 (pbest - x_i(t)) + b_2 r_2 (gbest - x_i(t)) \text{ (versión global)}$$

- ii. Evaluar la función de aptitud con las nuevas posiciones de las partículas.
- iii. Actualizar los mejores valores (pbest, lbest/gbest).

$r_1$  y  $r_2$  son valores aleatorios que toman valores entre 0 y 1



- ✓ El parámetro  $a$  llamado inercia, controla la fracción en el desplazamiento con la misma dirección que la partícula llevaba en el paso previo. Un valor de inercia alto presiona hacia la exploración global en una nueva área de búsqueda, mientras que un valor pequeño presiona hacia el ajuste en el área de búsqueda actual. Se sugieren valores entre  $[0.4, 0.9]$  [5].
- ✓ El parámetro  $b_1$  controla la fracción del desplazamiento en dirección de la mejor solución encontrada por la partícula.
- ✓ El parámetro  $b_2$  controla la fracción del desplazamiento en dirección de la mejor solución encontrada por el vecindario (versión local) o por el enjambre (versión global). Se sugiere que la suma  $b_1 + b_2 = 4$  [5].

## Bibliografía

1. Rozenberg, G., Bäck, T., & Kok, J.N. (2012). Handbook of Natural Computing (1st. ed.). Springer Publishing Company, Incorporated.
2. S. Olariu, A. Y. Zomaya. 2006. Handbook of Bioinspired Algorithms and Applications. Chapman and Hall/CRC, ISBN 978-1-58488-475-0.
3. Floreano, D., & Mattiussi, C. (2008). Bio-inspired artificial intelligence: theories, methods, and technologies. MIT press.
4. Slowik, A. (Ed.). (2020). Swarm Intelligence Algorithms: A Tutorial.
5. Keller, J.M., Liu, D., & Fogel, D.B. (2016). Fundamentals of computational intelligence: neural networks, fuzzy systems, and evolutionary computation. John Wiley & Sons.
6. Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization. Proceedings of ICNN'95 International Conference on Neural Networks, vol. 4, pp. 1942-1948.





¡ Gracias !

Thanks !

Obrigado

Xie xie ni

Domo arigatou

Спасибо

Merci

Grazie

Alfa Beta