UNIVERSIDAD NACIONAL GUILLERMO BROWN

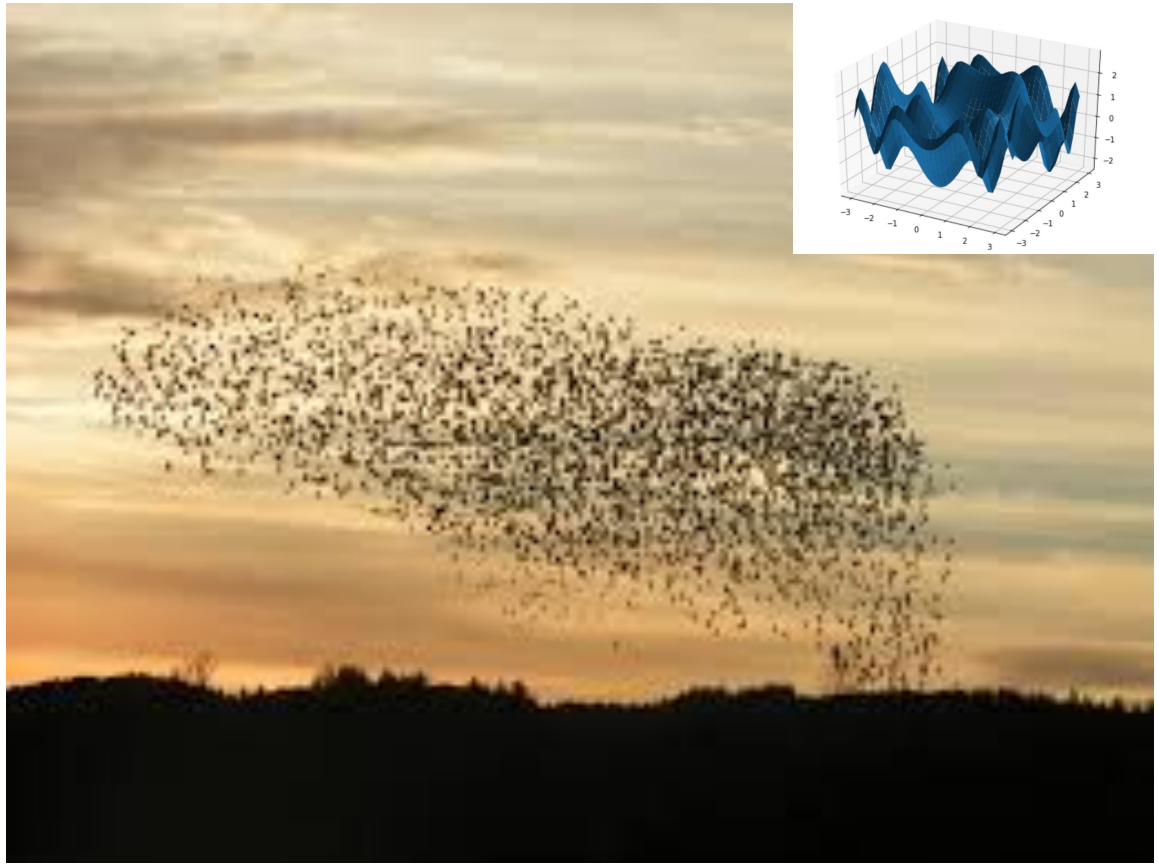
Licenciatura en

CIENCIA DE DATOS

Tecnicatura en

PROGRAMACIÓN

ALGORITMOS DE ENJAMBRE



Este algoritmo tiene una historia de colaboración entre distintas especialidades profesionales que lo hace más interesante que por haber sido uno de los pilares de la inteligencia artificial por casi una década.

Y un ejemplo de que cosas muy técnicas salen de profesiones que a primera vista nos parecen más artísticas que técnicas.

Particle Swarm Optimization

James Kennedy[1] and Russell Eberhart[2]

[1]Washington, DC 20212
kennedy_jim@bls.gov

[2]Purdue School of Engineering and Technology
Indianapolis, IN 46202-5160
eberhart@engr.iupui.edu

ABSTRACT

A concept for the optimization of nonlinear functions using particle swarm methodology is introduced. The evolution of several paradigms is outlined, and an implementation of one of the paradigms is discussed. Benchmark testing of the paradigm is described, and applications, including nonlinear function optimization and neural network training, are proposed. The relationships between particle swarm optimization and both artificial life and genetic algorithms are described.

1 INTRODUCTION

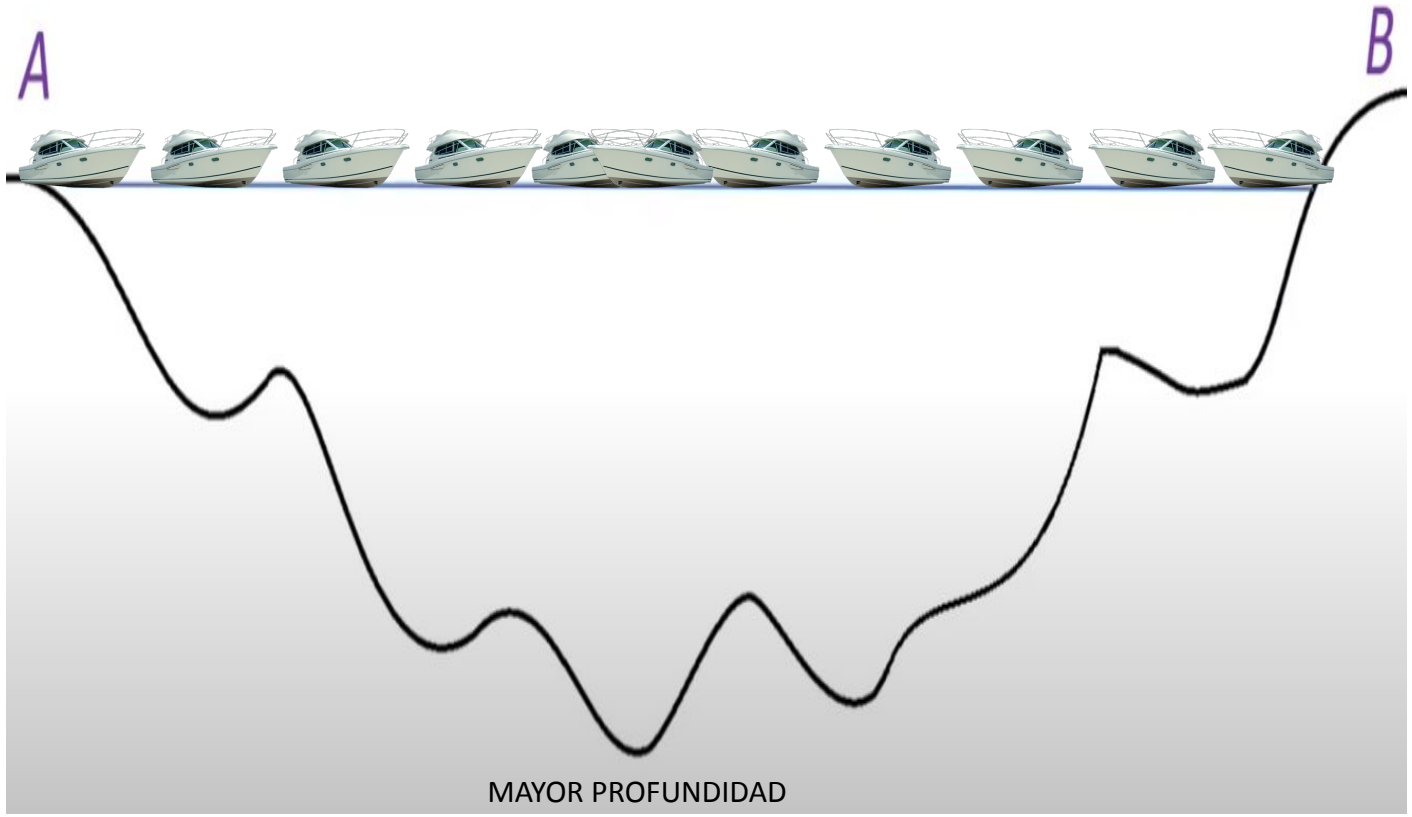
ALGORITMOS DE ENJAMBRE

Se basa en el comportamiento colaborativo de varios agentes independientes.
En la naturaleza se observa en el comportamiento de enjambres de insectos, bandadas de aves o cardúmenes de peces.

Estos comportamientos colaborativos les permiten escapar de depredadores y encontrar de manera muy eficiente las mejores fuentes de alimentación

VERSION SIMPLE

Se emplean dos barcos para determinar la profundidad máxima de un canal.
Los barcos tienen sondas que miden la profundidad debajo de ellos.
Comienzan en los extremos del canal.
El barco que mide la mayor profundidad "atrae" al que mide la menor, mientras permanece quieto.
En cada paso, los barcos se acercan al punto más profundo y se deberían encontrar ambos allí.

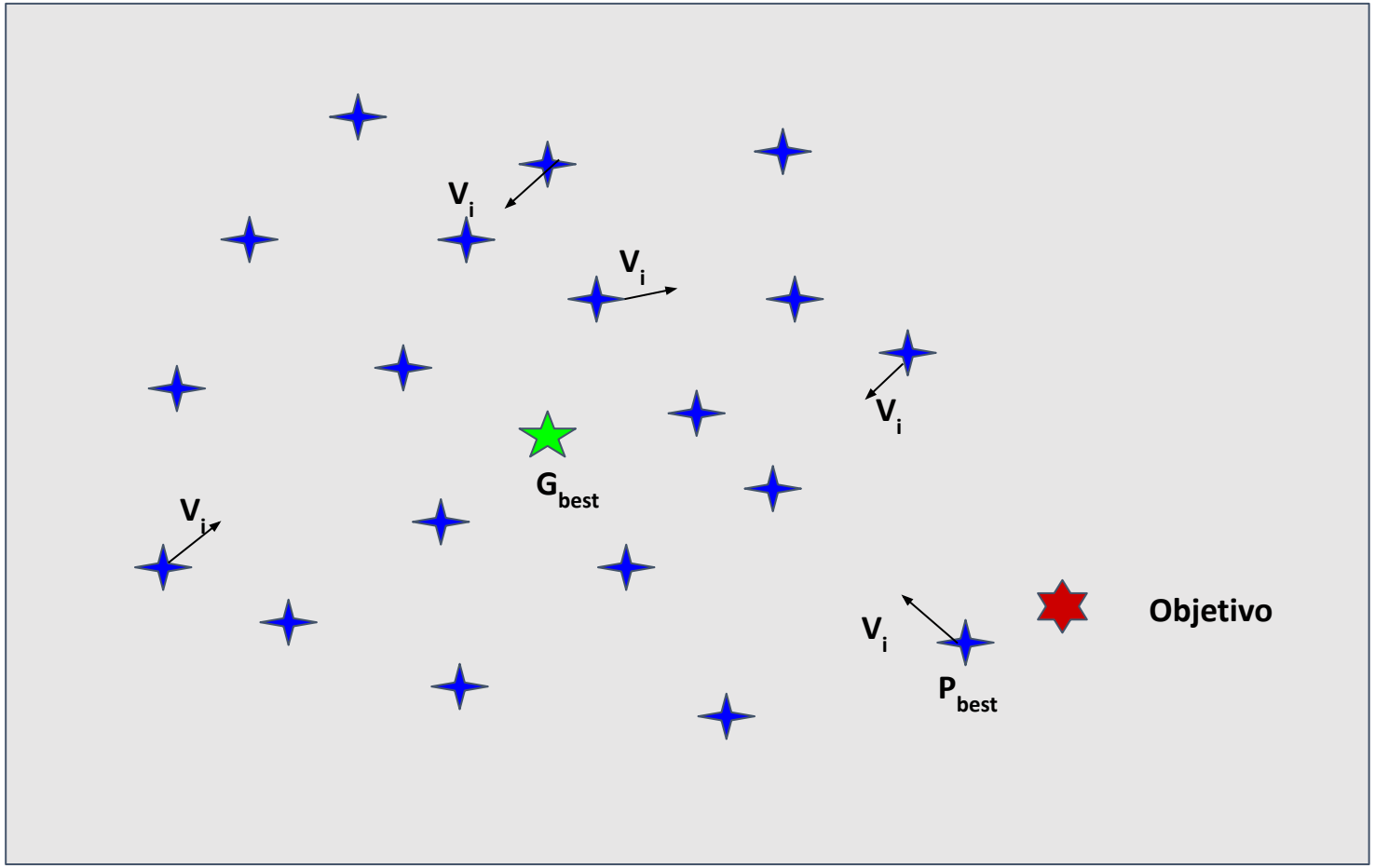


VERSION COMPLEJA

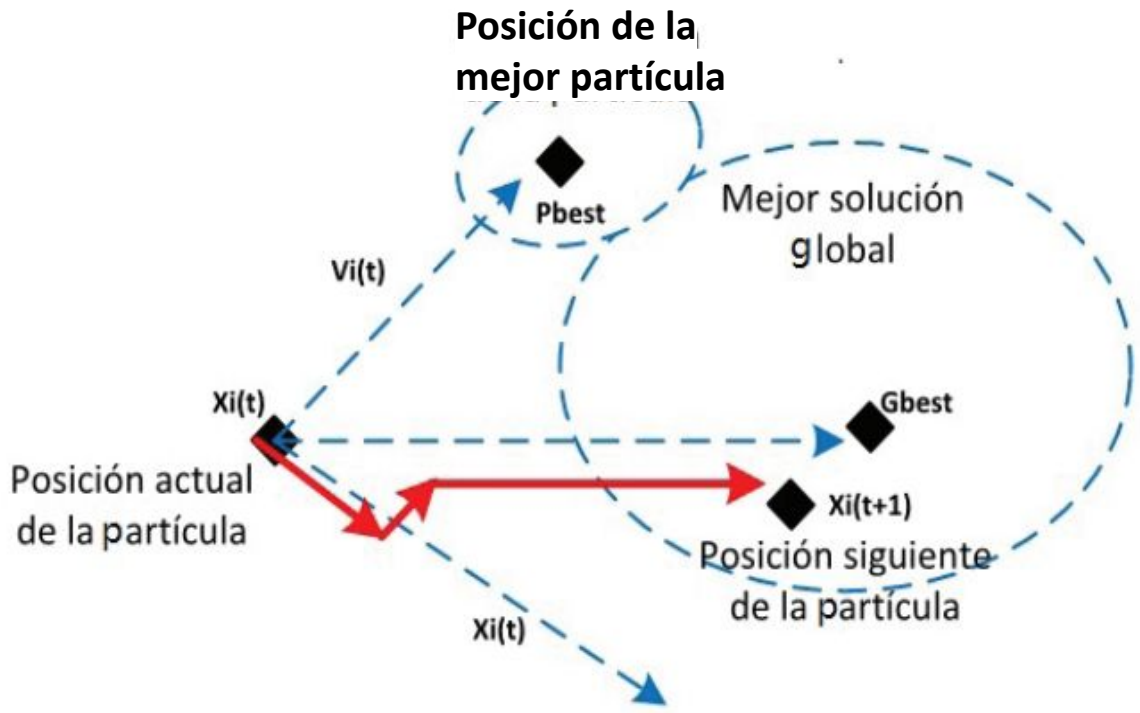
Se tiene un enjambre de partículas y un objetivo.
Todas las partículas se mueven relativamente al azar con una V_i pero mantienen la permanencia en el enjambre.

Una de las partículas es la más cercana al objetivo, esta es P_{best}

El centro de masa del enjambre "atrae" a las partículas hacia el, este es el G_{best}



VERSION COMPLEJA



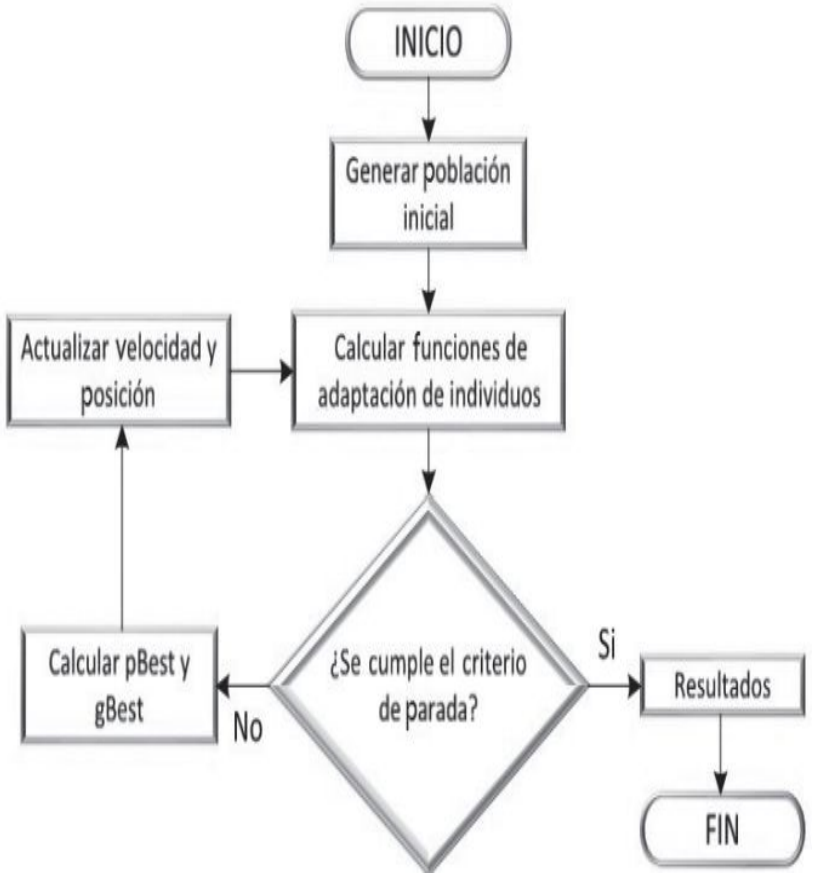
La segunda versión del método de optimización por enjambre adiciona la información del individuo más apto, de la posición global del enjambre y de la "inercia" que tiene cada individuo del enjambre.
De esta manera el comportamiento "más azaroso" de cada individuo y del global del enjambre aporta una información que acelera la detección del punto óptimo.

DESARROLLO DEL ALGORITMO

Al igual que con el algoritmo genético, necesitamos una función Fitness.
Esta función nos dirá cuán cerca o cuán lejos estamos del objetivo buscado.
En el caso de los animales, puede ser cuan fuerte huelen la comida.

El algoritmo tiene los siguientes pasos:

1 Generar las partículas en posiciones y direcciones al azar
2 Calcular la posición promedio del enjambre G_{best}
3 Encontrar el individuo más óptimo (funcion Fitness) P_{best}
4 Si el Fitness del P_{best} es ideal o se supero el numero de iteraciones, detener el algoritmo.
5 Si no, calcular la nueva posición de todas las partículas y volver al punto 2



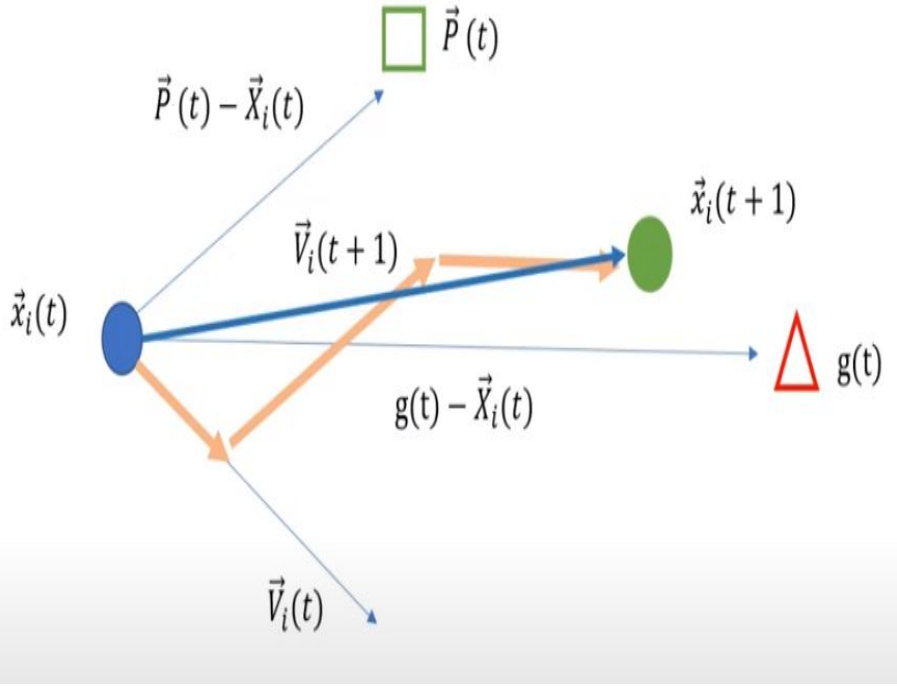
ACTUALIZACIÓN DE LA POSICIÓN

Al igual que con los AG, el problema más importante a la hora de implementar AE es la definición de la función fitness.
Esa función está muy relacionada con el problema que buscamos resolver.
Pero, normalmente es una función matemática de varias variables (coordenadas, peso, medidas de tamaño, etc)

El otro trabajo a realizar es el de calcular la nueva posición de la partícula.
Cada partícula pasa de una posición $X_i(t)$ a $X_i(t+1)$ en cada paso del algoritmo.

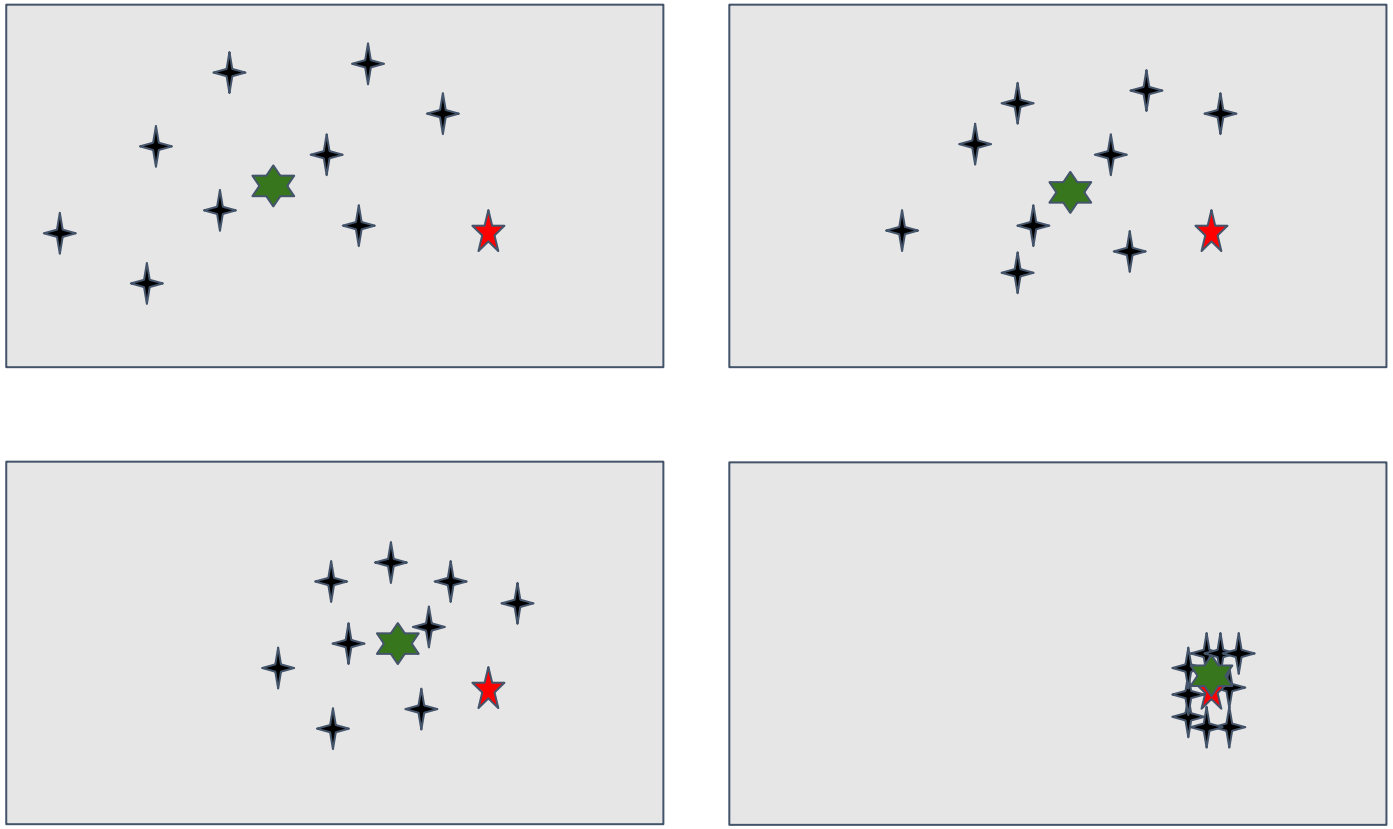
$$X_i(t+1) = X_i(t) + V_i(t+1)$$

Y V_i es el "vector" que calculamos sumando los vectores hacia P_{best}, G_{best} y la Velocidad inicial de la partícula (P , g(t) y V_i en la figura)



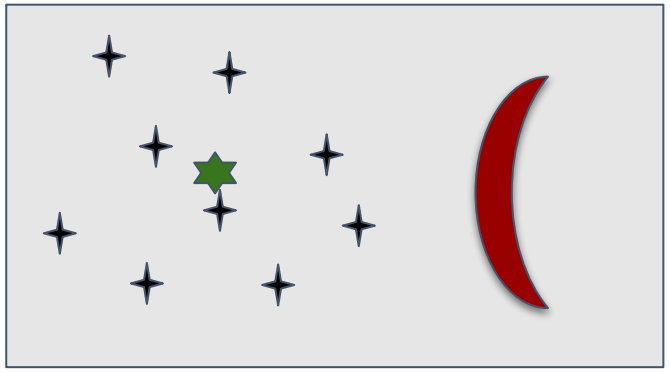
ACTUALIZACIÓN DE LA POSICIÓN DE TODO EL ENJAMBRE

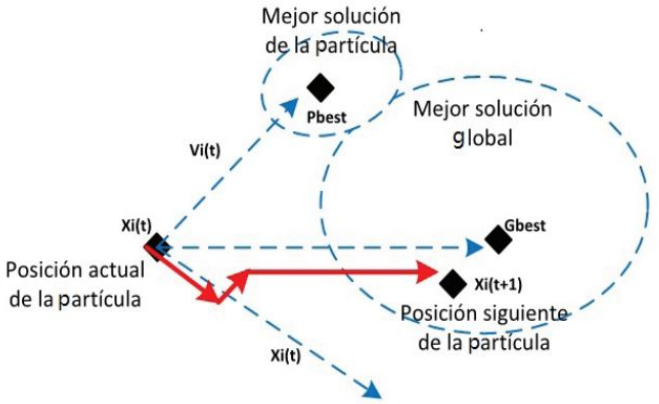
Este proceso de recálculo de la posición se realiza sobre todas las partículas del enjambre.
Entonces, a cada iteracion, el conjunto total de partículas se termina acercando al objetivo, aunque algunas partículas (por efecto de su posición relativa al centro del enjambre, su distancia al punto objetivo o a su inercia previa) inicialmente se ALEJEN.



OBJETIVOS MULTIPLES O DISTRIBUIDO

Hay ocasiones donde el objetivo no está concentrado en una posición, si no que ocupa varios lugares o está extendido en el espacio.
Un ejemplo podría ser el de un curso de agua para pájaros sedientos.
En estos casos CADA partícula tiene un P_{best}. O sea no es único para el individuo mejor ubicado.
El cálculo de la siguiente posición para cada partícula sigue siendo igual, solo que ahora hay un P_i para cada partícula X_i





HIPER-PARAMETROS

La ecuación de la nueva posición es:

$$X_i(t+1) = X_i(t) + P_i(t) + G_i(t)$$

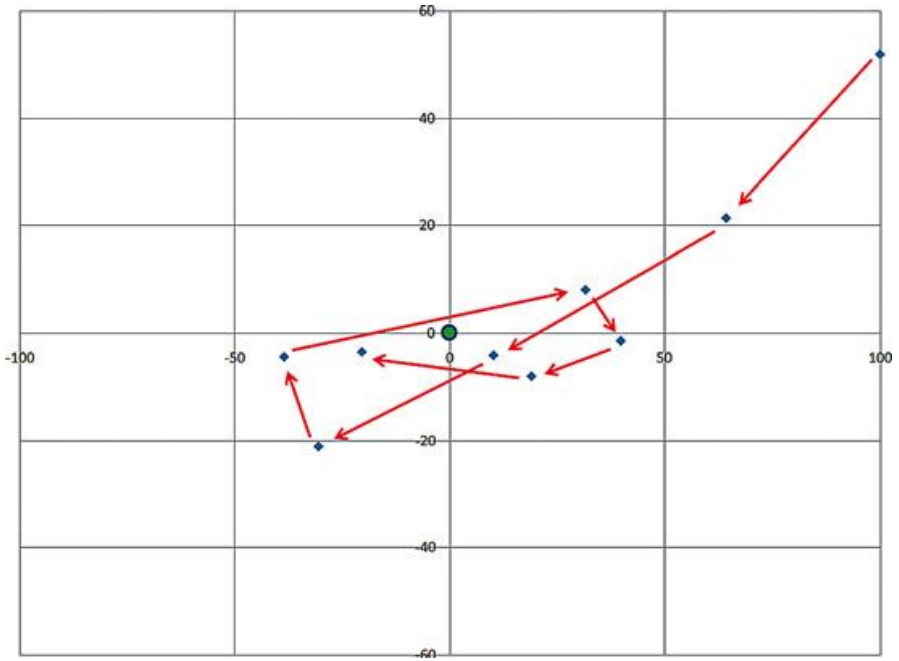
Pero a cada miembro se lo multiplica por una constante que depende del problema (o tambien deben ser determinados)

$$X_i(t+1) = WX_i(t) + C_1P_i(t) + C_2G_i(t)$$

A W, C_1 y C_2 se los denomina HIPER PARÁMETROS del algoritmo.

Y pueden determinar el rendimiento del algoritmo, si son muy grandes se produce la "búsqueda espiral"

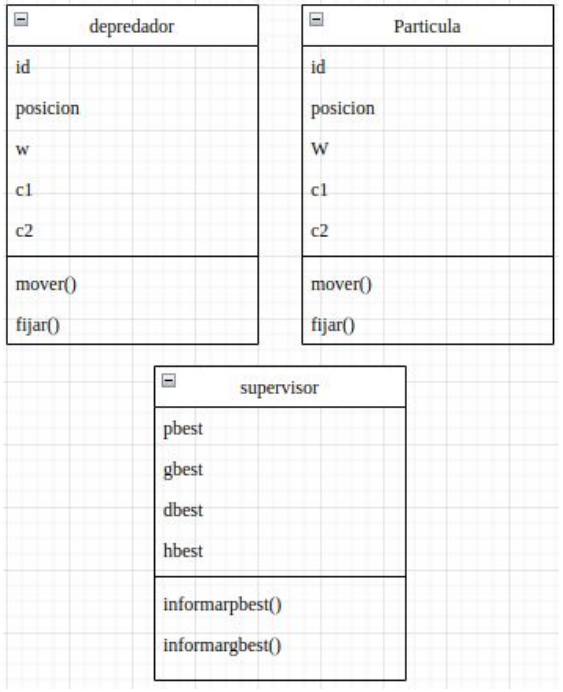
También son importantes en otra variación del algoritmo



DEPREDADORES/PRESAS

La última versión de los algoritmos de enjambre que conoceremos es una en la que el objetivo también es una partícula. O varias particulas.
Y esta partícula interacciona con el enjambre.
El enjambre puede perseguirla (PRESA) o huir de ella (DEPREDADOR).

Incluso puede haber interacciones complejas donde se huya de algunas y se persiga a otras al mismo tiempo.



Se considera a cada partícula como un **objeto** con atributos y métodos. Entre esos atributos están los HIPERPARAMETROS y los métodos de movimiento de la partícula dependen de los hiperparametros propios y los de las demás partículas.

A su vez todos ellos van recibiendo mensajes enviado por un objeto supervisor (controlador) que es el que determina cuál partícula está mejor ubicada y cual es la posición global del enjambre.

VENTAJAS Y DESVENTAJAS DEL ALGORITMO

Ventajas:
Es altamente parallelizable, sacando provecho de las arquitecturas concurrentes.
Es rapido, un orden de magnitud mejor que el GA equivalente.
Es repetible, una vez conocida la función fitness es seguir una receta.
Se puede aplicar a problemas muy complejos
Aprobado por la naturaleza

Desventajas:
No siempre es sencillo encontrar la función fitness.
Es muy sensible a pequeños cambios en los hiper-parámetros