Inteligencia Social

Optimización basada en organizaciones sociales biológicas

M. E. Mazzei

UNA

Organización de los Insectos Sociales

- Qué los gobierna?
- Cómo siguen órdenes?
- Cómo se planifican?
- Cómo preservan el equilibrio?

Enjambre

Qué es?: En biología se define un enjambre, como un conjunto o una agrupación de organismos semejantes, que permanecen juntos y se mueven en una misma dirección.

Enjambre

Son ejemplos de enjambres:

- Colonias de termitas
- Bandadas de pájaros para encontrar alimentos
- Colonias de abejas para reproducirse
- Un sistema inmune es un enjambre de células y moléculas
- Una multitud de personas

Características de las organizaciones de Insectos Sociales

- Siguen reglas simples. Muestran conductas complejas que surgen entre individuos que exhiben una conducta simple (**auto-organización**).
- Poseen autonomía: Las interacciones conducen a una conducta global inteligente.
- Poseen funcionamiento distribuido.

Conceptos

La **auto-organización**, proporciona poderosas herramientas para transferir conocimiento.

Una colonia de hormigas resuelve problemas de:

- Búsqueda de alimentos
- Construcción de los nidos
- Alimentación de las crías
- Distribución del trabajo entre individuos

$\longrightarrow$ Tiene sus contrapartes en las ciencias de la computación y en ingeniería.

Conceptos

- La idea más importante es usar los principios de auto-organización de las sociedades de insectos para coordinar poblaciones de agentes artificiales que colaboran para resolver problemas computacionales.
- Una de las características más resaltantes que se toma de estas sociedades es la adaptación a los cambios de ambiente.
- Interacción entre los miembros de la sociedad
- Estimergía: Comunicación indirecta.

¿Qué es Inteligencia de Enjambres?

- En este contexto es el diseño y la utilización de algoritmos o técnicas inspiradas en la conducta colectiva de las colonias de insectos u otras sociedades de animales.
- Un enjambre es una población de soluciones que interactúan entre si y es capaz de optimizar una función objetivo a través de una búsqueda colaborativa en un espacio.

Sistemas de Enjambres Artificiales

Se trata de diseñar sistemas artificiales con el uso del computador, que sean:

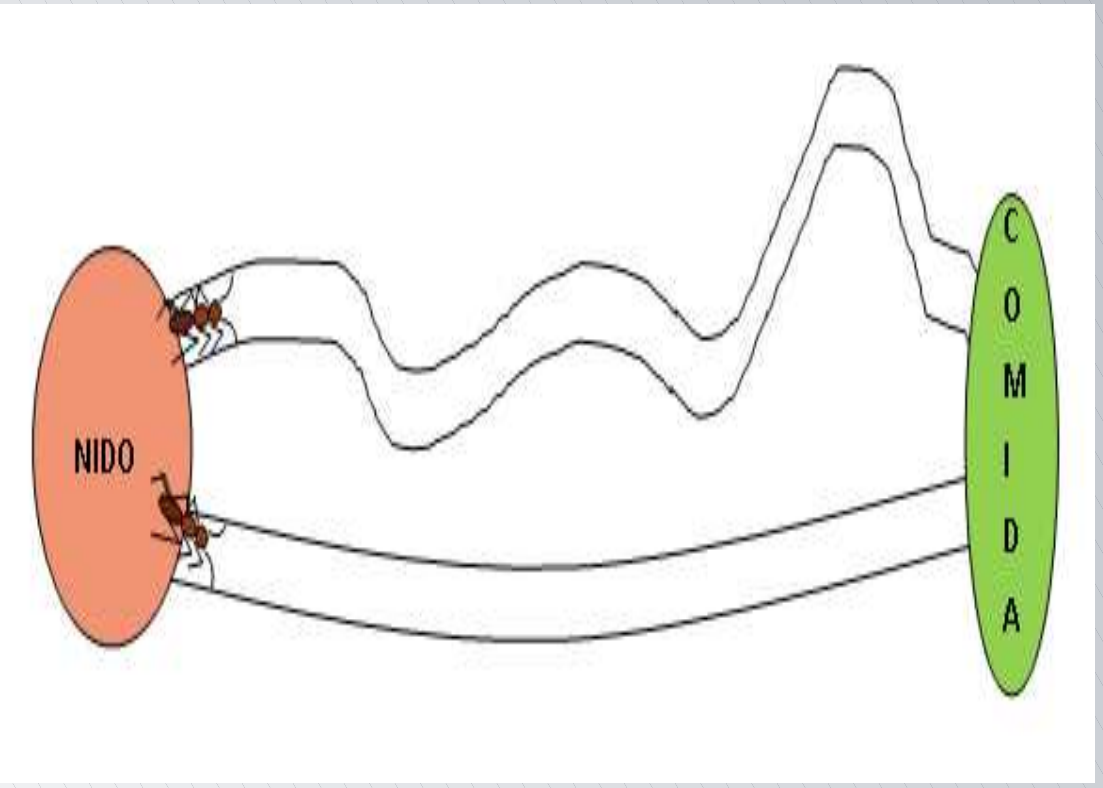
- Adaptativos
- Descentralizados
- Flexibles
- Robustos y capaces de resolver problemas.

Sistemas de Enjambres Artificiales

Trataremos dos técnicas heurísticas:

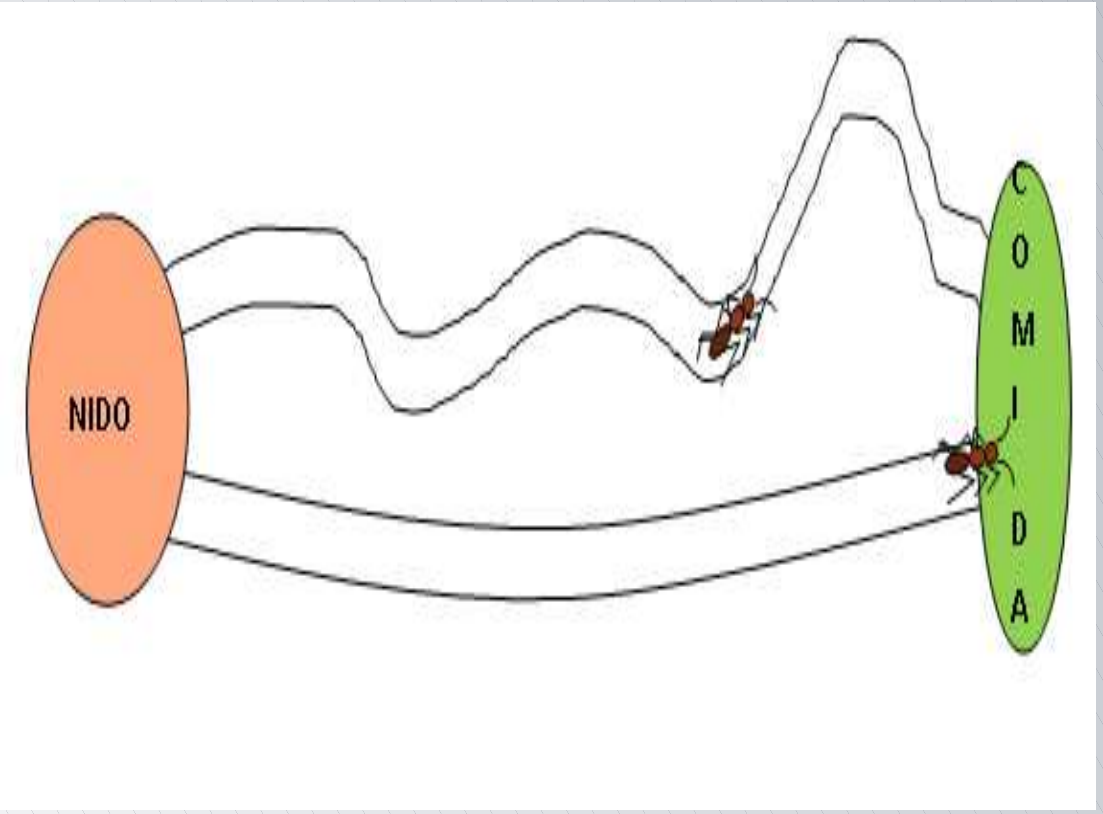
- Algoritmos Hormiga
- Enjambres de partículas

Colonias de Hormigas



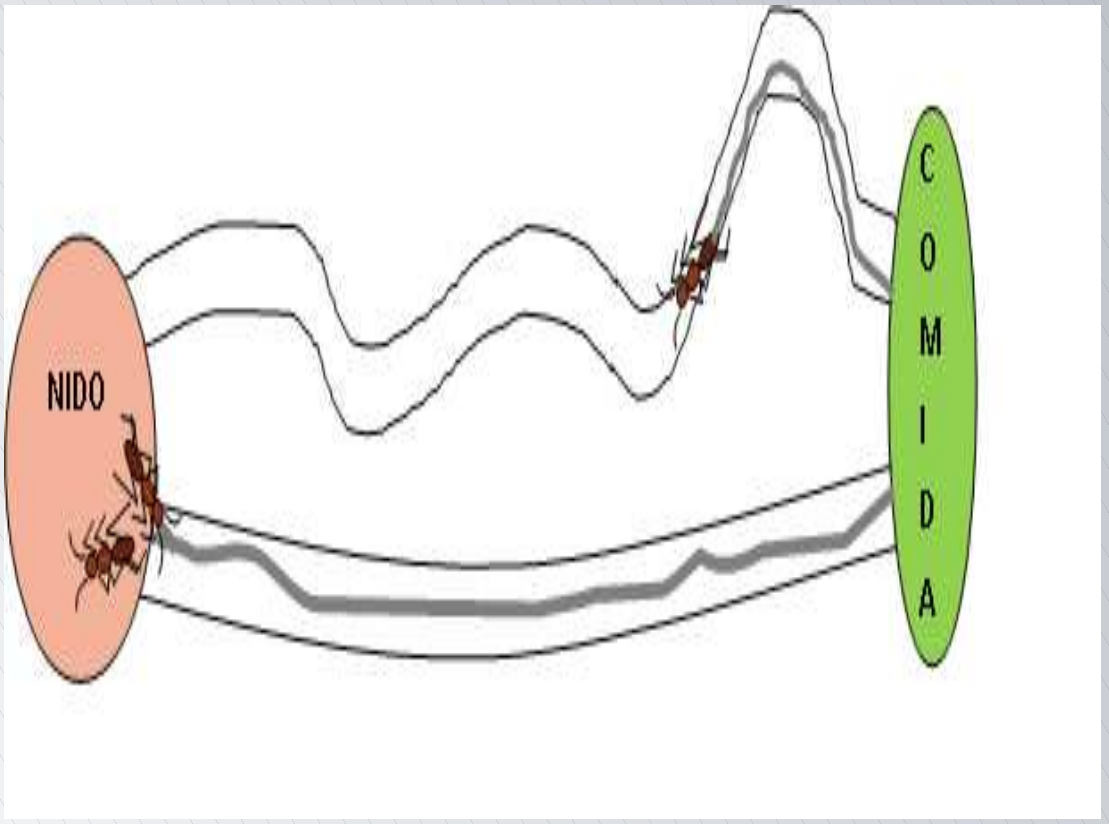
Cómo actúan las hormigas en la búsqueda de alimento

Colonias de Hormigas



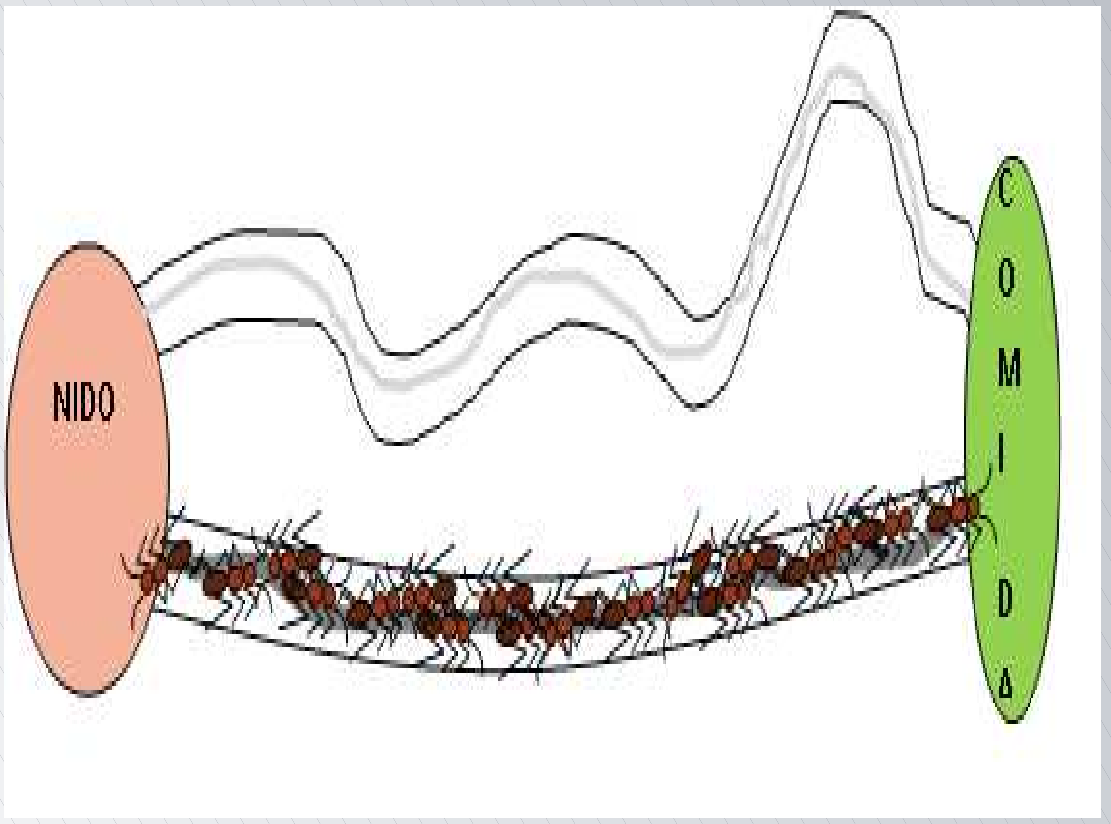
Cómo actúan las hormigas en la búsqueda de alimento

Colonias de Hormigas



Cómo actúan las hormigas en la búsqueda de alimento?

Colonias de Hormigas



Cómo actúan las hormigas en la búsqueda de alimento

Algoritmos Basados en Colonias de Hormigas(ACO)

- Los algoritmos hormiga se inspiran en la organización de las hormigas para construir las soluciones. Fueron desarrollados por M. Dorigo en 1992.
- Las soluciones se construyen de manera incremental.
- Están basados en heurísticas(experiencias, exploración)
- Pueden aplicarse en cualquier problema de optimización combinatoria.

Algoritmos ACO y el Problema del Agente Viajero

- Las hormigas artificiales construyen soluciones haciendo caminatas aleatorias en un grafo conexo.
- Cada hormiga es colocada inicialmente en una ciudad al azar.
- En cada iteración, escoge una ciudad que no ha visitado, empleando la transición probabilística.
- En cada iteración se actualizan los rastros de feromonas.
- La construcción de la solución termina cuando se han visitado todas las ciudades.

Algoritmos ACO y el Problema del Agente Viajero

Rastro de feromona en arco (i,j): $\tau_{ij}(t)$

Visibilidad: $\eta_{ij} = 1/d_{ij}$

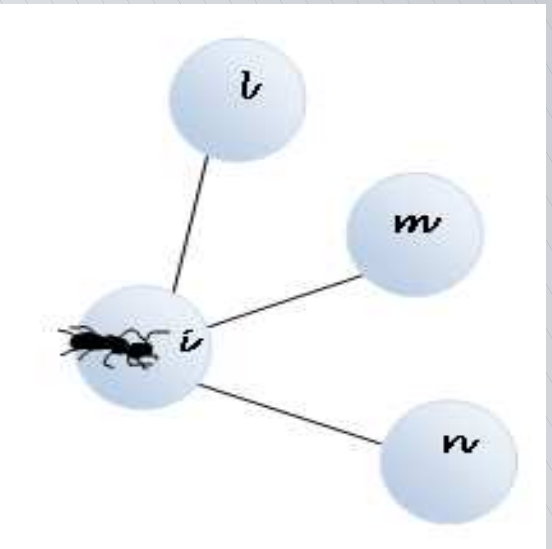
$$p_{ic}(t) = \frac{[\varphi_{ic}(t)]^\alpha}{[\varphi_{ic}(t)]^\alpha + [\varphi_{il}(t)]^\alpha}, \quad p_{il}(t) = \frac{[\varphi_{il}(t)]^\alpha}{[\varphi_{ic}(t)]^\alpha + [\varphi_{il}(t)]^\alpha}$$

α y β son parámetros, α ajusta el rastro de feromona y β la visibilidad

Algoritmos ACO y el Problema del Agente Viajero

Transición:

$$p_{ic}(t) = \frac{[\varphi_{ic}(t)]^{\alpha}}{[\varphi_{ic}(t)]^{\alpha}+[\varphi_{il}(t)]^{\alpha}}, \quad p_{il}(t) = \frac{[\varphi_{il}(t)]^{\alpha}}{[\varphi_{ic}(t)]^{\alpha}+[\varphi_{il}(t)]^{\alpha}}$$



La hormiga debe decidir a cuál ciudad visitar

Algoritmo en seudolenguaje ..

Comienzo

Para todas las k hormigas

Generar aleatoriamente k posiciones.

Colocar las hormigas en las posiciones, $\tau(0) = \tau_0$

Fin Para

Para t = 1 hasta tmax hacer

Para todas las k hormigas

Construir una ruta Tk(t), calcular:

$$p_{ij}^k(t) = \frac{[\varphi_{ic}(t)]^\alpha}{[\varphi_{ic}(t)]^\alpha + [\varphi_{il}(t)]^\alpha}, \quad p_{il}(t) = \frac{[\varphi_{il}(t)]^\alpha}{[\varphi_{ic}(t)]^\alpha + [\varphi_{il}(t)]^\alpha}$$

Fin Para

Si se ha encontrado una ruta mejor, entonces actualizar T+ y L+ (la ruta y su longitud).

continúa ...

Algoritmo en seudolenguaje ..

Para cada arco (i, j) hacer

Actualizar feromonas: $\tau_{ij} \leftarrow (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t) + e\tau^{e}_{ij}(t)$, donde

$$\Delta\tau_{ij}(t) = \sum_{i=1}^{n} \Delta\tau^{k}_{ij}(t)$$

$$\Delta\tau^{k}_{ij} = \begin{cases} \frac{1}{C^k}, & \text{si arco } (i,j) \text{ pertenece a } T^k \\ 0, & \text{en otro caso} \end{cases} \quad (1)$$

Fin Para

Para todo arco (i, j) hacer

$$\tau_{ij}(t+1) = \tau_{ij}(t)$$

Fin Para

Fin Para

Fin

Aplicaciones de los algoritmos Hormiga

Se emplean en la resolución de problemas de combinatoria, entre ellos:

- Problemas de rutas (redes estáticas)
- Problemas de redes dinámicas
- Problemas de asignación
- Problemas de planificación de horarios
- Problemas de subconjuntos
- Otros más recientes como los de aprendizaje de máquinas(*machine learning*), a través de los cuales se pueden resolver problemas de clasificación de patrones.
- También se han combinado estos algoritmos con otras técnicas para resolver problemas, estas se conocen como **híbridas**.

Optimización por Enjambres de Partículas

Optimización por Enjambres de Partículas

Los algoritmos de Optimización de Partículas o algoritmos PSO (Particle Swarm Optimization) fueron desarrollados por Kennedy y Eberhart en 1995.

Se trata de métodos de optimización para funciones no lineales en espacios continuos y discretos, basados en el concepto de bandadas de aves y cardúmenes.

PSO

Los enjambres de individuos se mueven libremente en un espacio, en búsqueda de alimento. Cuando un individuo lo halla, de alguna manera se comunica con sus vecinos, por lo que el enjambre seguirá a dicho individuo.

El término partícula se refiere a elementos de la población que no poseen masa ni volumen pero si poseen movimiento, cuando se les puede llamar "puntos"se trata de elementos que permanecen estáticos. Se le asocia a cada partícula una posición y una velocidad.

PSO

Para simular este tipo de agrupación, en el computador, se emplean 3 reglas muy simples:

1. Evitar las colisiones de individuos con sus vecinos.
2. Ajustar la velocidad de un individuo a la misma de sus vecinos.
3. Hacer que cada individuo permanezca cerca de sus vecinos.

PSO

Las partículas cambian de estado al desplazarse dentro del espacio de búsqueda. Cada partícula i es una solución y se trata como un punto del espacio, se le representa como:

$$x = (x_1, \ldots, x_d)$$

Se evalúa el çomportamiento"de cada partícula, de acuerdo a una función de costo o de *fitness*. La que tiene un menor costo asociado (problema de minimización) se denominará:

$$p = (p_1, \ldots, p_d)$$

PSO

La tasa de cambio es la velocidad, y se denota por:

$$v = (v_1, \ldots, v_d)$$

La velocidad de cada partícula se calcula de acuerdo a la siguiente ecuación:

$$v_j^{(k+1)} = v_j^{(k)} + c_1\epsilon_1(p_j^{(k)} - x_j^{(i)}) + c_2\epsilon_2(g_j^{(k)} - x_j^{(k)}) \tag{2}$$

$$x_j^{(k+1)} = x_j^{(k)} + v_j^{(k+1)} \tag{3}$$

ϵ_1 y ϵ_2 son números aleatorios uniformes en (0,1]

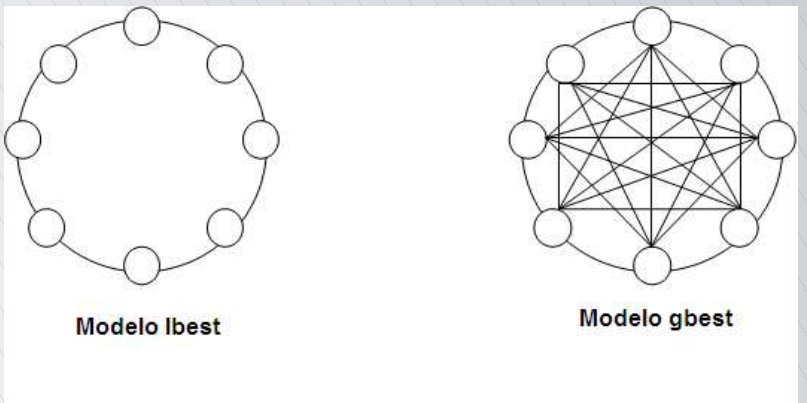
PSO

El algoritmo considera una memoria de partículas. En la ecuación de actualización de la velocidad, p representa la mejor partícula i en su historia y g la mejor partícula en su vecindario. Existen diferentes maneras de tratar la organización de partículas.

La **topología** de las partículas es su forma de agrupación y relación.

PSO

Las topologías más conocidas son la **anillo** y la **estrella**. Existen otras como: la Von Newmann y cualquier otra con la que sea posible experimentar. Originalmente se denominó al modelo en el que se empleaba la topología anillo, modelo *lbest* y el asociado a la topología estrella, modelo *gbest*



PSO

El algoritmo genera una población inicial de partículas con posición y velocidad al azar. Estas partículas cambian de posición y velocidad de acuerdo al desempeño del grupo.

Es importante considerar el tipo de topología a emplear, en cuanto a la relación de las mismas con sus vecinas. El proceso se puede realizar un número fijo de iteraciones(maxiter) o también se puede comparar dos iterados sucesivos y dependiendo de su aproximación el proceso se detiene o continúa.

Algoritmo PSO

Comienzo

Crear población inicial de partículas $\{x^i\}$

Para cada partícula i en el enjambre hacer

Si $f(x) < f(p)$ entonces Copiar (p, x) // copia x en p

$g = i$ // asignación arbitraria

Para $j \in J$ //J *conjunto de índices de las partículas vecinas*

Si $f(p_j) < f(p_g)$ entonces $g = j$ //g *el índice de mejor en vecindad*

Fin Para

Para $d = 1$ hasta D hacer

$$v_d(t) = v_d(t-1) + c_1\epsilon_1(p_d - x_d(t-1)) + c_2\epsilon_2(g_d - x_d(t-1))$$

$$x_d = x_d + v_d$$

Fin Para

Fin Para

Fin

Ajustes y Variantes al algoritmo PSO

- Ajuste de la velocidad: Para evitar que la velocidad alcance valores muy grandes, lo cual ocasiona igualmente valores muy grandes para las posiciones en el espacio de búsqueda,se limita a un valor V_max, en mayor absoluto no mayor que 4
- La introducción del peso inicial w, frena el crecimiento de la velocidad, de manera dinámica, ocasionando en un principio una exploración global y al final un refinamiento de la solución.

$$v_d^{(i)} = wv_d^{(i)} + c_1\epsilon_1(p_d^{(i)} - x_d^{(i)}) + c_2\epsilon_2(g_d^{(i)} - x_d^{(i)}) \quad (4)$$

Ajustes y Variantes al algoritmo PSO

El peso inercial se actualiza de acuerdo a la siguiente ecuación:

$$w = w_{max} - (w_{max} - w_{min})\frac{k}{maxiter}$$

Comúnmente se emplea $w_{max} = 0,9$ y $w_{min} = 0,4$

- Empleo del factor de constricción χ, el cual afecta a todos los términos presentes en el cálculo de v_j

$$v_j^{(i)} = \chi[v_j^{(i)} + c_1\epsilon_1(p_j^{(i)} - x_j^{(i)}) + c_2\epsilon_2(g_j^{(i)} - x_j^{(i)})] \quad (5)$$

  en donde χ es: $\chi = \frac{2}{|2-\varphi-\sqrt{\varphi^2-4\varphi}|}, \quad \varphi = c_1 + c_2 = 4{,}1$

- Existen otros ajustes de parámetros. Algunos han sido propuestos por Clerc M., para evitar el estancamiento de las soluciones.

Resumen PSO

La técnica heurística del PSO, se emplea para la optimización de funciones no lineales en variable continua, en casos de: funciones sin restricciones, funciones con restricciones y multi-objetivo.

Por lo tanto se puede aplicar en la resolución de todo tipo de problemas que presenten esta estructura. La ventaja del PSO radica en que las funciones no necesitan ser continuas ni diferenciables.