BIOINFORMÁTICA
2013 - 2014

PARTE I. INTRODUCCIÓN

- Tema 1. Computación Basada en Modelos Naturales

PARTE II. MODELOS BASADOS EN ADAPTACIÓN SOCIAL (Swarm Intelligence)

- Tema 2. Introducción a los Modelos Basados en Adaptación Social
- Tema 3. Optimización Basada en Colonias de Hormigas
- Tema 4. Optimización Basada en Nubes de Partículas (Particle Swarm)

PARTE III. COMPUTACÍON EVOLUTIVA

- Tema 5. Introducción a la Computación Evolutiva
- Tema 6. Algoritmos Genéticos I. Conceptos Básicos
- Tema 7. Algoritmos Genéticos II. Diversidad y Convergencia
- Tema 8. Algoritmos Genéticos III. Múltiples Soluciones en Problemas Multimodales
- Tema 9. Estrategias de Evolución y Programación Evolutiva
- Tema 10. Algoritmos Basados en Evolución Diferencial (Diferential Evolution – DE)
- Tema 11. Modelos de Evolución Basados en Estimación de Distribuciones (EDA)
- Tema 12. Algoritmos Evolutivos para Problemas Multiobjetivo
- Tema 13. Programación Genética
- Tema 14. Modelos Evolutivos de Aprendizaje

PARTE IV. OTROS MODELOS DE COMPUTACIÓN BIOINSPIRADOS

- Tema 15. Sistemas Inmunológicos Artificiales
- Tema 16. Otros Modelos de Computación Natural/Bioinspirados

BIOINFORMÁTICA

TEMA 4. OPTIMIZACIÓN BASADA EN NUBES DE PARTÍCULAS (PARTICLE SWARM)

1. INTRODUCCIÓN Y RÁPIDO RESUMEN
2. FUNCIONAMIENTO DEL ALGORITMO PSO
3. ASPECTOS AVANZADOS
4. APLICACIONES Y RECURSOS ELECTRÓNICOS

Kennedy, J., Eberhart, R.C. Swarm Intelligence. Morgan Kauffmann, 2001.

1. INTRODUCCIÓN Y RÁPIDO RESUMEN

- La "Particle Swarm Optimization" (PSO) es una metaheurística poblacional inspirada en el comportamiento social del vuelo de las bandadas de aves y el movimiento de los bancos de peces.

- La población se compone de varias partículas (nube de partículas = particle swarm) que se mueven (*"vuelan"*) por el espacio de búsqueda durante la ejecución del algoritmo.

- Este movimiento de cada partícula *p* depende de:
  - Su mejor posición desde que comenzó el algoritmo (*pBest*),
  - la mejor posición de las partículas de su entorno (*lBest*) o de toda la nube (*gBest*) desde que comenzó el algoritmo.

  En cada iteración, se cambia aleatoriamente la velocidad de *p* para acercarla a las posiciones *pBest* y *lBest/gBest*.

1. INTRODUCCIÓN Y RÁPIDO RESUMEN (2)

- Desarrollo: USA, en 1995.
- Primeros autores: Russ C. Eberhart y James Kennedy

  Kennedy, J. and Eberhart, R. (1995). "Particle Swarm Optimization", Proc. 1995 IEEE Intl. Conf. on Neural Networks, pp. 1942-1948, IEEE Press.
- Aplicación típica:
  - Optimización continua (optimización de parámetros reales, numérica).
- Características atribuidas:
  - Asume un intercambio de información (interacciones sociales) entre los agentes de búsqueda.
  - Idea básica: guardar información del mejor propio y global.
  - Implementación muy sencilla, pocos parámetros.
  - Convergencia rápida a buenas soluciones.

2. FUNCIONAMIENTO DEL ALGORITMO PSO

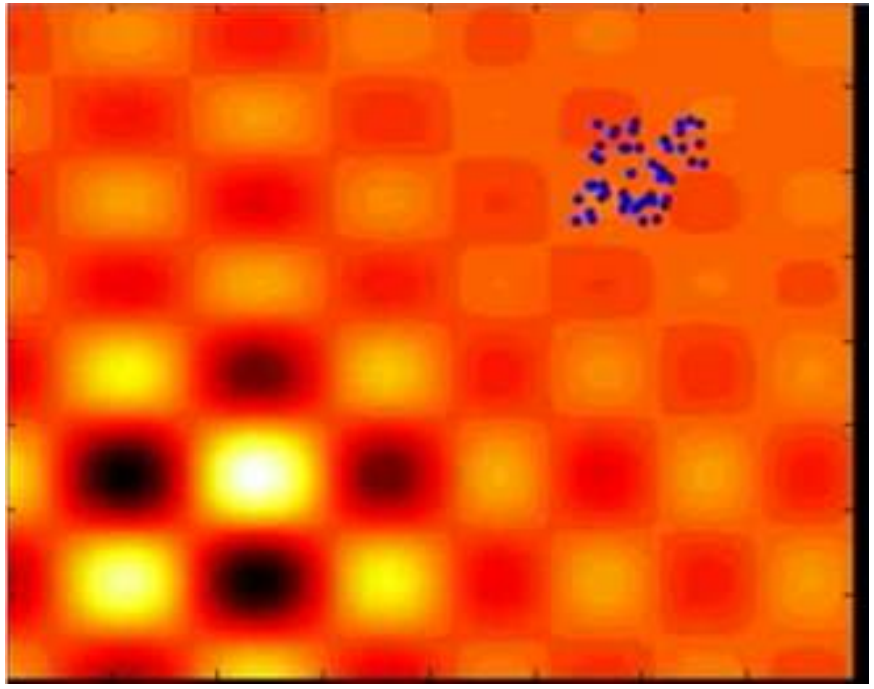
- FUNCIONAMIENTO BÁSICO
- ANATOMÍA DE UNA PARTÍCULA
- INICIALIZACIÓN DE LA NUBE DE PARTÍCULAS
- MOVIMIENTO DE LAS PARTÍCULAS
- PSEUDOCÓDIGOS
- VALORES DE LOS PARÁMETROS
- TOPOLOGÍAS DE LA NUBE DE PARTÍCULAS

Funcionamiento Básico

- PSO simula el comportamiento de las bandadas de aves.
- Supongamos que una de estas bandadas busca comida en un área y que solamente hay una pieza de comida en dicha área.
- Los pájaros no saben donde está la comida pero sí conocen su distancia a la misma.
- La estrategia más eficaz para hallar la comida es seguir al ave que se encuentre más cerca de ella.

Funcionamiento Básico (2)

- PSO emula este escenario para resolver problemas de optimización. Cada solución (partícula) es un *"ave"* en el espacio de búsqueda que está siempre en continuo movimiento y que nunca muere.



Funcionamiento Básico (2)

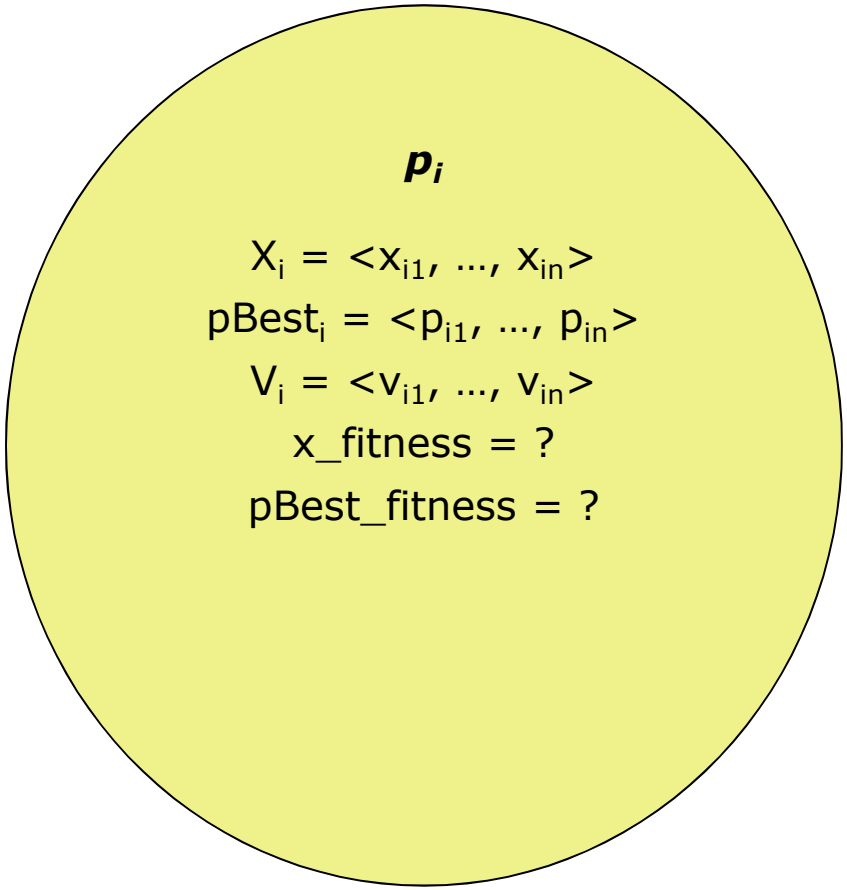
- La nube de partículas es un sistema multiagente. Las partículas son agentes simples que se mueven por el espacio de búsqueda y que guardan (y posiblemente comunican) la mejor solución que han encontrado.

- Cada partícula tiene un fitness, una posición y un vector velocidad que dirige su *"vuelo"*. El movimiento de las partículas por el espacio está guiado por las partículas óptimas en el momento actual.

Anatomía de una Partícula

Una partícula está compuesta por:

- Tres vectores:
  - El **vector X** almacena la posición actual (localización) de la partícula en el espacio de búsqueda,
  - El **vector pBest** almacena la localización de la mejor solución encontrada por la partícula hasta el moemento, y
  - El **vector V** almacena el gradiente (dirección) según el cuál se moverá la partícula.
- Dos valores de fitness:
  - El **x_fitness** almacena el fitness de la solución actual (vector X), y
  - El **p_fitness** almacena el fitness de la mejor solución local (vector *pBest*).



Inicialización de la Nube de Partículas

- La nube se inicializa generando las posiciones y las velocidades iniciales de las partículas.

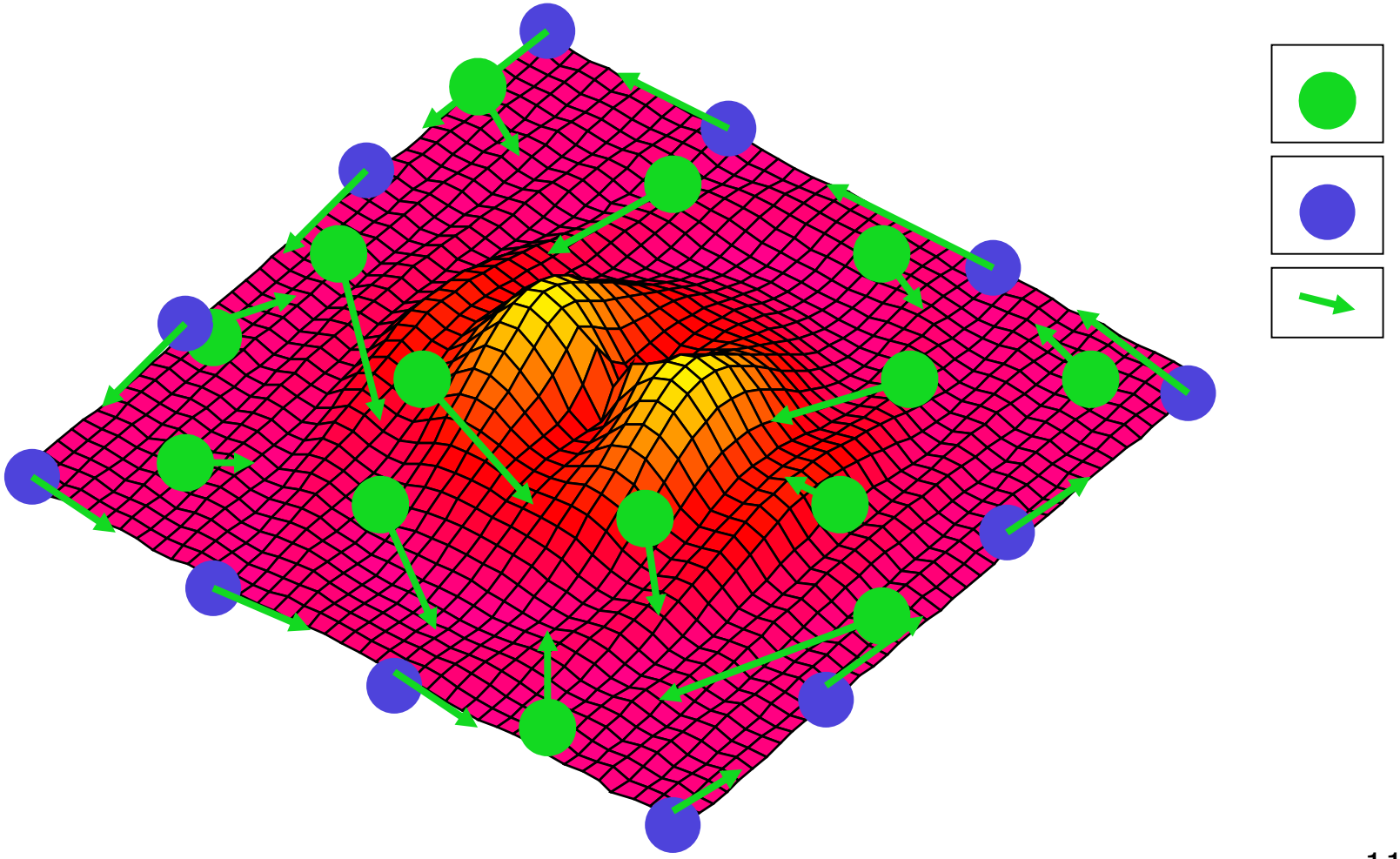
- Las posiciones se pueden generar aleatoriamente en el espacio de búsqueda, de forma regular, o con una combinación de ambas.

- Las velocidades se generan aleatoriamente, con cada componente en el intervalo $[-V_{max}, V_{max}]$.

  No es conveniente fijarlas a cero, no se obtienen buenos resultados.

  V_{max} será la velocidad máxima que pueda tomar una partícula en cada movimiento.

Inicialización de la Nube de Partículas (2)



Movimiento de las Partículas

¿Cómo se mueve una partícula de una posición del espacio de búsqueda a otra?

- Se hace simplemente añadiendo el vector velocidad V_i al vector posición X_i para obtener un nuevo vector posición:

$$X_i \leftarrow X_i + V_i$$

- Una vez calculada la nueva posición de la partícula, se evalúa ésta. Si el nuevo fitness es mejor que el que la partícula tenía hasta ahora, *pBest*_fitness, entonces:

$$pBest_i \leftarrow X_i \quad ; \quad pBest\_fitness \leftarrow x\_fitness.$$

Movimiento de las Partículas (2)

- De este modo, el primer paso es ajustar el vector velocidad, para después sumárselo al vector posición.

- Las fórmulas empleadas son las siguientes:

$$v_{id} = v_{id} + \varphi_1 \cdot rnd() \cdot (pBest_{id} - x_{id}) + \varphi_2 \cdot rnd() \cdot (g_{id} - x_{id})$$

COGNITIVO **SOCIAL**

$$x_{id} = x_{id} + v_{id}$$

donde:

  - **p_i** es la partícula en cuestión, $pBest_{id}$ es la mejor solución encontrada por la partícula.
  - φ_1, φ_2 son ratios de aprendizaje (pesos) que controlan los componentes **cognitivo** y **social**,
  - **g** representa el índice de la partícula con el mejor *pBest*_fitness del entorno de p_i (*lBest*) o de toda la nube (*gBest*),
  - los **rnd()** son números aleatorios generados en [0,1], y
  - **d** es la d-ésima dimensión del vector.

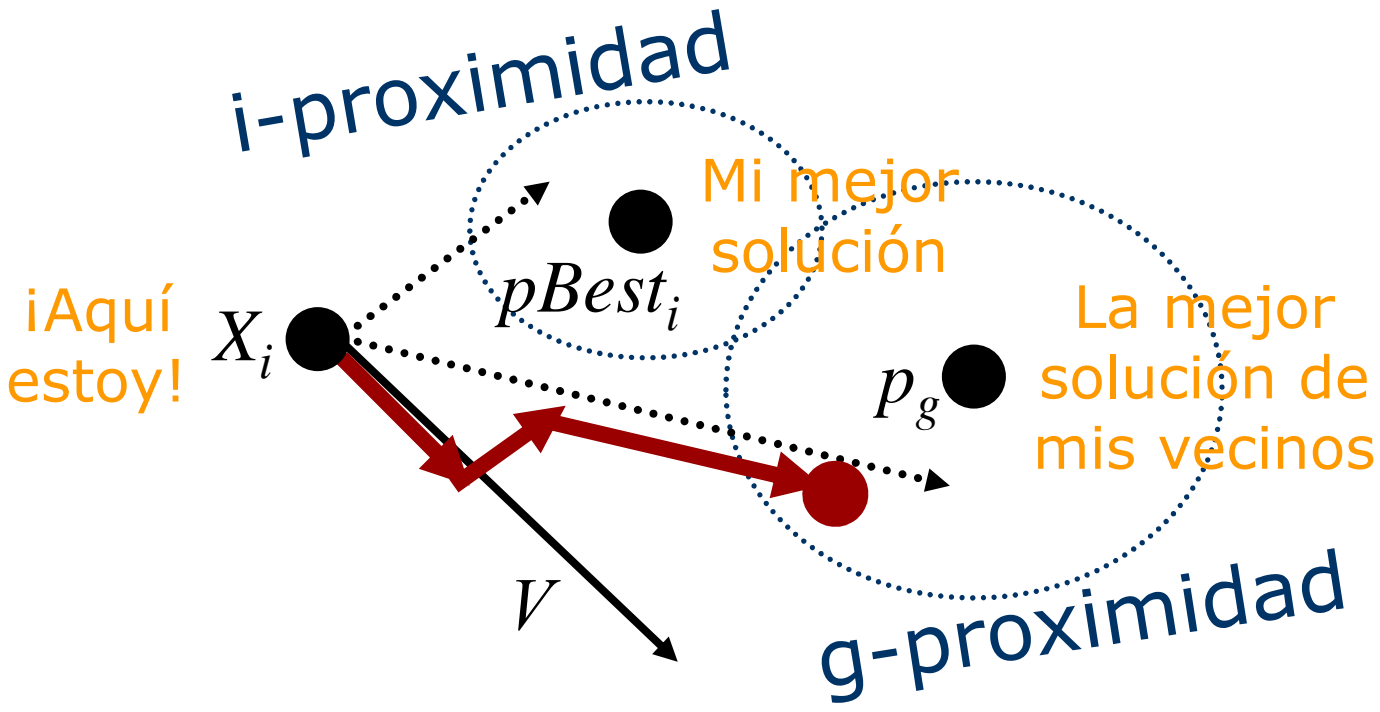
Movimiento de las Partículas (3)

TIPOS DE ALGORITMOS DE PSO:

- Kennedy identifica cuatro tipos de algoritmos de PSO en función de los valores de φ_1 y φ_2:
  - Modelo completo: $\varphi_1, \varphi_2 > 0$.
  - Sólo Cognitivo: $\varphi_1 > 0$ y $\varphi_2 = 0$.
  - Sólo Social: $\varphi_1 = 0$ y $\varphi_2 > 0$.
  - Sólo Social exclusivo: $\varphi_1 = 0$, $\varphi_2 > 0$ y $g \neq i$ (la partícula en sí no puede ser la mejor de su entorno).

Movimiento de las Partículas (4)

REPRESENTACIÓN GRÁFICA:



Pseudocódigo PSO Local

t = 0;
Para i=1 hasta Número_partículas
    inicializar X_i y V_i;

Mientras (no se cumpla la condición de parada) hacer
    t ← t + 1
    Para i=1 hasta Número_partículas
        evaluar X_i;
        Si $F(X_i)$ es mejor que $F(pBest)$ entonces
            $pBest_i \leftarrow X_i$; $F(pBest_i) \leftarrow F(X_i)$
    Para i=1 hasta Número_partículas
        Escoger $lBest_i$, la partícula con mejor fitness del entorno de X_i
        Calcular V_i, la velocidad de X_i, de acuerdo a $pBest_i$ y $lBest_i$
        Calcular la nueva posición X_i, de acuerdo a X_i y V_i

Devolver la mejor solución encontrada

Pseudocódigo PSO Global

t = 0;
Para i=1 hasta Número_partículas
    inicializar X_i y V_i;

Mientras (no se cumpla la condición de parada) hacer
    t ← t + 1
    Para i=1 hasta Número_partículas
        evaluar X_i;
        Si F(X_i) es mejor que F(*pBest*) entonces
            $pBest_i \leftarrow X_i$; F($pBest_i$) ← F(X_i)
        Si F(*pBest*) es mejor que F(*gBest*) entonces
            $gBest \leftarrow pBest_i$; F($gBest_i$) ← F($pBest_i$)

    Para i=1 hasta Número_partículas
        Calcular V_i, la velocidad de X_i, de acuerdo a $pBest_i$ y ***$gBest_i$***
        Calcular la nueva posición X_i, de acuerdo a X_i y V_i

Devolver la mejor solución encontrada

Valores de los Parámetros

- Tamaño de la nube: Entre 20 y 40 partículas (problemas simples, 10; problemas muy complejos, 100-200).
- Velocidad máxima: V_{max} se suele definir a partir del intervalo de cada variable.
- Ratios de aprendizaje: Habitualmente, $\varphi_1=\varphi_2=2$.
- PSO Global vs. PSO Local: La versión global converge más rápido pero cae más fácilmente en óptimos locales y viceversa.

Topologías de la Nube de Partículas

- Las topologías definen el entorno de cada partícula individual. La propia partícula siempre pertenece a su entorno.

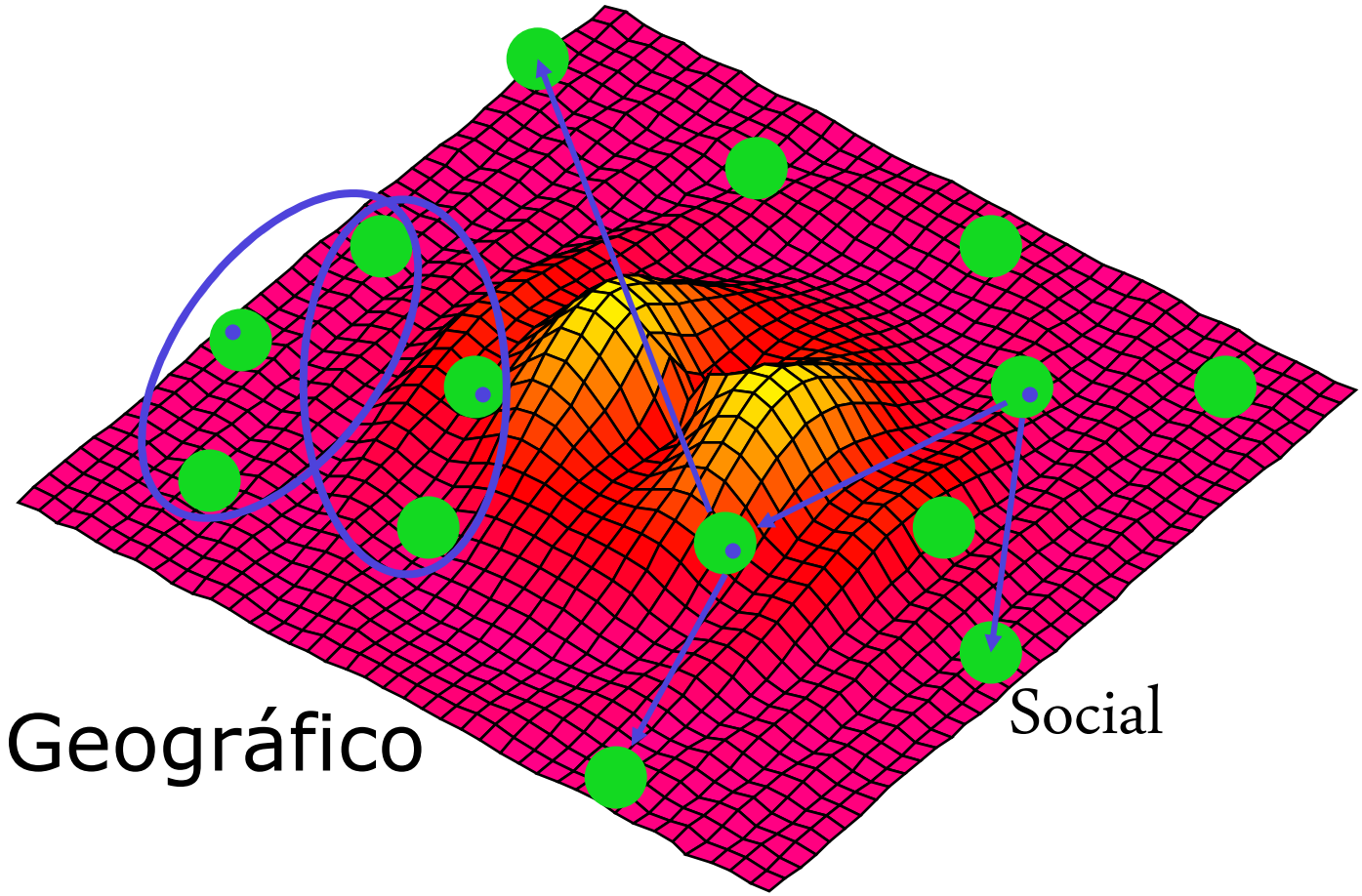
- Los entornos pueden ser de dos tipos:
  - Geográficos: se calcula la distancia de la partícula actual al resto y se toman las más cercanas para componer su entorno.
  - Sociales: se define a priori una lista de vecinas para partícula, independientemente de su posición en el espacio.

- Los entornos sociales son los más empleados.

- Una vez decidido el entorno, es necesario definir su tamaño. El algoritmo no es muy sensible a este parámetro (3 o 5 son valores habituales con buen comportamiento).

- Cuando el tamaño es toda la nube de partículas, el entorno es a la vez geográfico y social, y tenemos la PSO global.

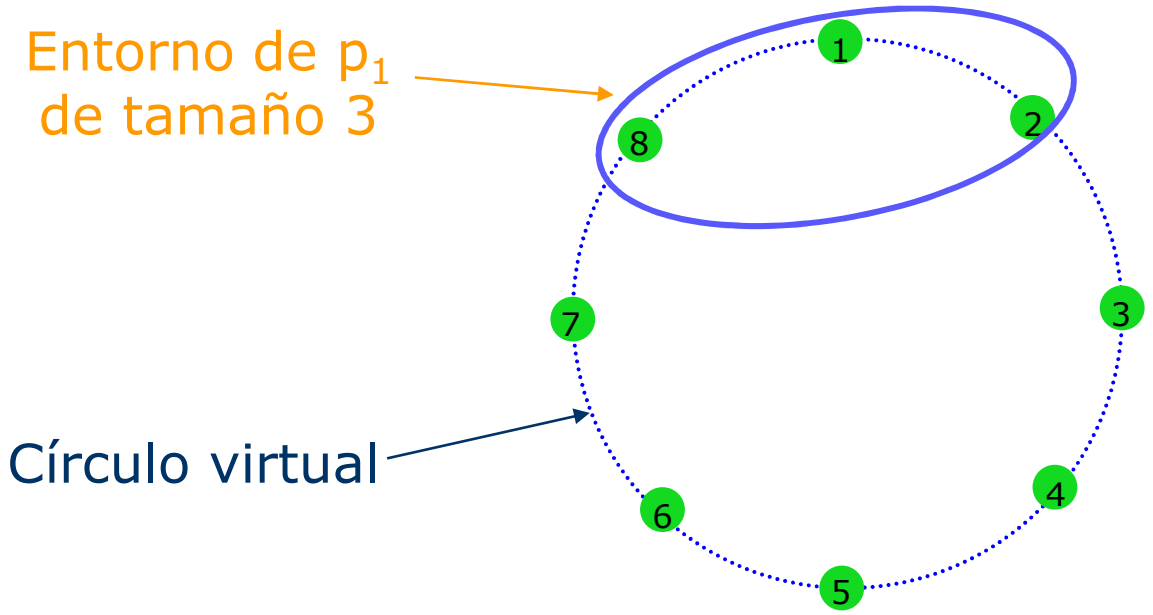
Topologías de la Nube de Partículas (2)



Topologías de la Nube de Partículas (3)

- La topología social más empleada es la de anillo, en la que se considera un vecindario circular.

- Se numera cada partícula, se construye un círculo virtual con estos números y se define el entorno de una partícula con sus vecinas en el círculo:



3. ASPECTOS AVANZADOS

- **CONTROL DE LA VELOCIDAD DE LAS PARTÍCULAS**
- **TAMAÑO DE LA NUBE DE PARTÍCULAS**
- **INFLUENCIA DEL TIPO DE ENTORNO**
- **ACTUALIZACIÓN DE LAS PARTÍCULAS**
- **ELECCIÓN DE VALORES ADAPTATIVOS PARA φ_1 Y φ_2**

Control de la Velocidad de las Partículas

- Un problema habitual de los algoritmos de PSO es que la magnitud de la velocidad suele llegar a ser muy grande durante la ejecución, con lo que las partículas se mueven demasiado rápido por el espacio.

- El rendimiento puede disminuir si no se fija adecuadamente el valor de V_{max}, la velocidad máxima inicial de cada componente del vector velocidad.

- Se han propuesto dos métodos para controlar el excesivo crecimiento de las velocidades:
  - Un factor de inercia, ajustado dinámicamente, y
  - Un coeficiente de constricción.

Control de la Velocidad de las Partículas (2) Factor de Inercia

- En este caso, la ecuación de adaptación de la velocidad pasa a ser la siguiente:

$$v_{id} = \omega \cdot v_{id} + \varphi_1 \cdot rnd() \cdot (pBest_{id} - x_{id}) + \varphi_2 \cdot rnd() \cdot (lBest_{id} - x_{id})$$

donde ω se inicializa a 1.0 y se va reduciendo gradualmente a lo largo del tiempo (medido en iteraciones del algoritmo).

- ω debe mantenerse entre 0.9 y 1.2. Valores altos provocan una búsqueda global (más diversificación) y valores bajos una búsqueda más localizada (mas intensificación).

Control de la Velocidad de las Partículas (3) Coeficiente de Constricción

- De nuevo, se realiza una modificación en la ecuación de adaptación, la siguiente:

$$v_{id} = K \cdot [v_{id} + \varphi_1 \cdot rnd() \cdot (pBest_{id} - x_{id}) + \varphi_2 \cdot rnd() \cdot (lBest_{id} - x_{id})]$$

donde:

- $K = \dfrac{2}{\left| \varphi - 2 + \sqrt{\varphi^2 - 4\varphi} \right|}$
- $\varphi = \varphi_1 + \varphi_2$
- $\varphi > 4$ (normalmente $\varphi = 4.1$, $\varphi_1 = \varphi_2$)

Tamaño de la Nube de Partículas

- El tamaño de la nube de partículas determina el equilibro entre la calidad de las soluciones obtenidas y el coste computacional (número de evaluaciones necesarias).

- Hace poco, se han propuesto algunas variantes que adaptan heurísticamente el tamaño de la nube:
  - Si la calidad del entorno de la partícula ha mejorado pero la partícula es la peor de su entorno, se elimina la partícula.
  - Si la partícula es la mejor de su entorno pero no hay mejora en el mismo, se crea una nueva partícula a partir de ella.

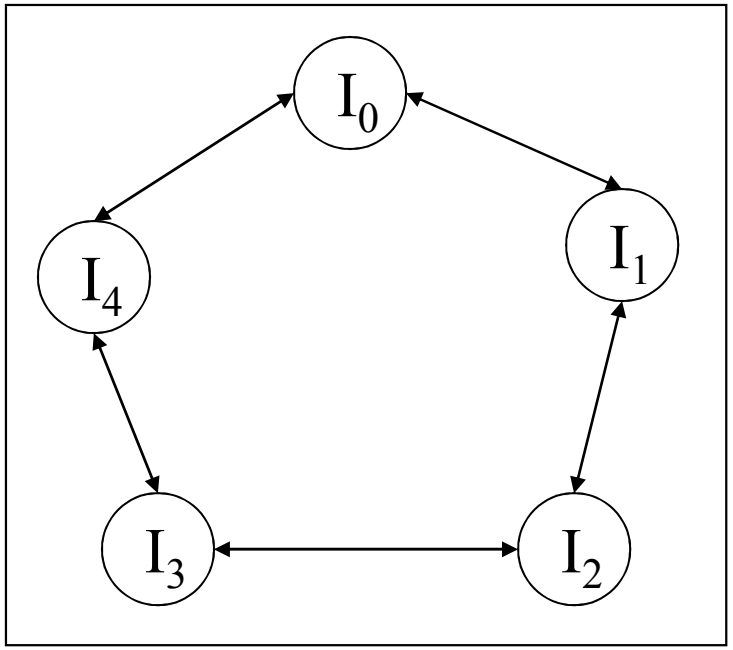
- Las decisiones se toman de forma probabilística en función del tamaño actual de la nube.

Influencia del Tipo de Entorno

- Los entornos globales parecen obtener mejores resultados desde el punto de vista del coste computacional.
- El rendimiento es similar a la topología de anillo y al del uso de entornos con tamaño mayor que 3.
- Se ha investigado poco en los efectos de la topología de la nube en el comportamiento de la búsqueda del algoritmo.
- Por otro lado, el tamaño del vecindario también se puede adaptar con la misma heurística del tamaño de la nube.

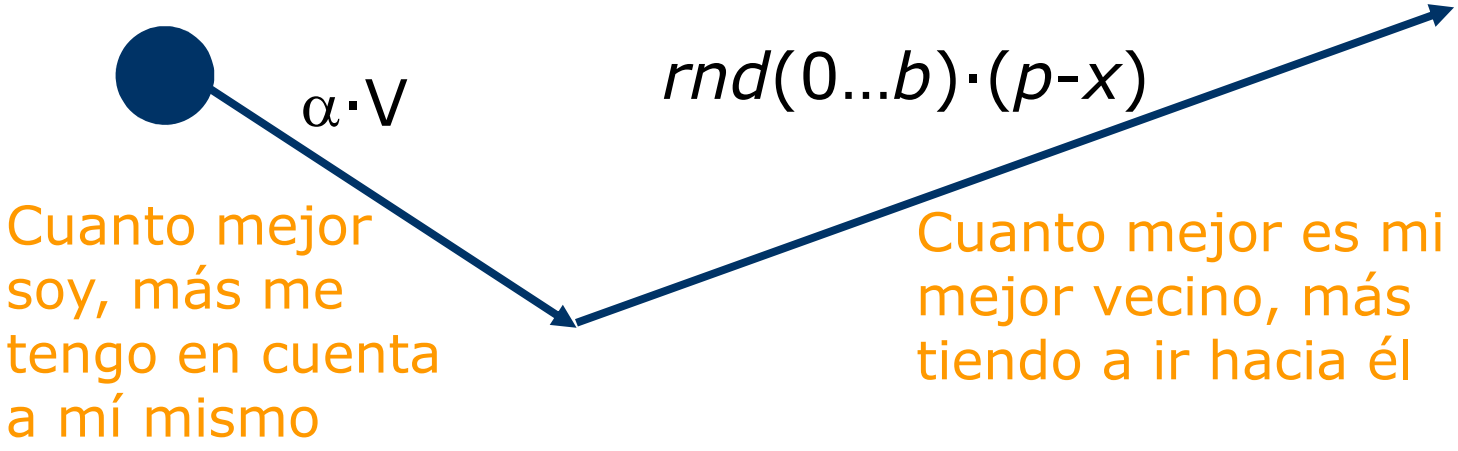
Actualización de las Partículas

- La actualización de las partículas se puede efectuar de dos formas distintas:
  - Síncrona
  - Asíncrona
- La actualización asíncrona permite considerar las soluciones nuevas más rápidamente.
- El método asíncrono puede representarse por el gráfico siguiente.



Elección de Valores Adaptativos para φ_1 y φ_2

- Los pesos que definen la importancia de las componentes cognitiva y social pueden definirse dinámicamente según la calidad de la propia partícula y del entorno:



4. APLICACIONES Y RECURSOS ELECTRÓNICOS

- Optimización de funciones numéricas.
- Entrenamiento de Redes Neuronales.
- Aprendizaje de Sistemas Difusos.
- Registrado de Imágenes.
- Viajante de Comercio.
- Control de Sistemas.
- Ingeniería Química.
- …

- Web site de PSO: http://www.swarmintelligence.org/

4. APLICACIONES Y RECURSOS ELECTRÓNICOS

- Versión discreta de PSO:

IEEE Transactions on Evolutionary Computation, 2010, V. 14:2, 278 - 300

A Novel Set-Based Particle Swarm Optimization Method for Discrete Optimization Problems

Chen, W.-N. Zhang, J. Chung, H. S. H. Zhong, W.-L. Wu, W.-G. Shi, Y.-h.
Digital Object Identifier : 10.1109/TEVC.2009.2030331
Online paper

4. APLICACIONES Y RECURSOS ELECTRÓNICOS

- Versiones recientes

Frankestein PSO: MA. Montes de Oca, T. Stützle, M. Birattari, M. Dorigo, Frankenstein's PSO: A Composite Particle Swarm Optimization Algorithm IEEE Transactions on Evolutionary Computation, Vol 13:5 (2009) pp. 1120-1132

OLPSO: Z-H Zhan, J. Zhang, Y. Li, Y-H. Shi, Orthogonal Learning Particle Swarm Optimization, IEEE Transactions on Evolutionary Computation Vol 15:6 pp. 832-847 (2011)

Implementación y artículo disponible en:

http://sci2s.ugr.es/EAMHCO/#Software

BIOINFORMÁTICA
2013 - 2014

PARTE I. INTRODUCCIÓN

- **Tema 1. Computación Basada en Modelos Naturales**

PARTE II. MODELOS BASADOS EN ADAPTACIÓN SOCIAL (Swarm Intelligence)

- **Tema 2. Introducción a los Modelos Basados en Adaptación Social**
- **Tema 3. Optimización Basada en Colonias de Hormigas**
- **Tema 4. Optimización Basada en Nubes de Partículas (Particle Swarm)**

PARTE III. COMPUTACÍON EVOLUTIVA

- **Tema 5. Introducción a la Computación Evolutiva**
- **Tema 6. Algoritmos Genéticos I. Conceptos Básicos**
- **Tema 7. Algoritmos Genéticos II. Diversidad y Convergencia**
- **Tema 8. Algoritmos Genéticos III. Múltiples Soluciones en Problemas Multimodales**
- **Tema 9. Estrategias de Evolución y Programación Evolutiva**
- **Tema 10. Algoritmos Basados en Evolución Diferencial (Diferential Evolution – DE)**
- **Tema 11. Modelos de Evolución Basados en Estimación de Distribuciones (EDA)**
- **Tema 12. Algoritmos Evolutivos para Problemas Multiobjetivo**
- **Tema 13. Programación Genética**
- **Tema 14. Modelos Evolutivos de Aprendizaje**

PARTE IV. OTROS MODELOS DE COMPUTACIÓN BIOINSPIRADOS

- **Tema 15. Sistemas Inmunológicos Artificiales**
- **Tema 16. Otros Modelos de Computación Natural/Bioinspirados**