Enjambres de Partículas

Jose Aguilar

Generalidades

- En las colonias de insectos, **el comportamiento de enjambre** está muy presente, entendido como el comportamiento de la colonia, en las que sus miembros interactúan, con el fin de optimizar un objetivo global, pudiendo ser ellos muy simples o complejos.

- Inspirados en ese comportamiento, en la literatura se han venido proponiendo **modelos de enjambre**.

- Algunos aspectos interesantes de esos modelos, es que normalmente
  - **se basan en interacciones locales**, las cuales suelen ser indirectas (estigmergia),
  - exhiben un comportamiento **auto-organizativo.**

Generalidades

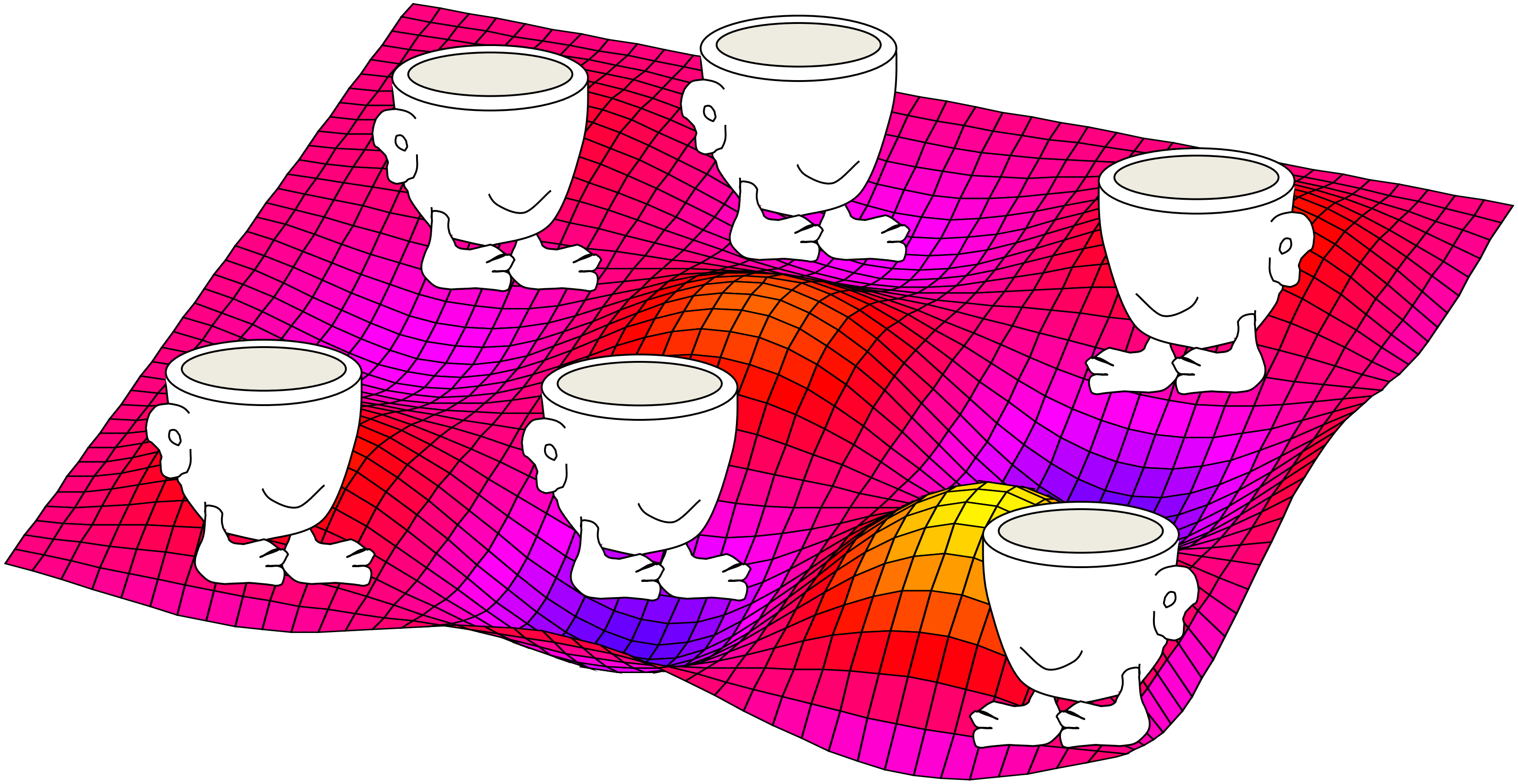
- En esta parte, más que caracterizar un proceso propio de un sistema de insecto en particular, presentamos un **modelo de optimización general, basado en un comportamiento colectivo**.

A ese modelo se le ha denominado en la literatura***, optimización por enjambre de partícula*** (PSO, por sus siglas en inglés) ,

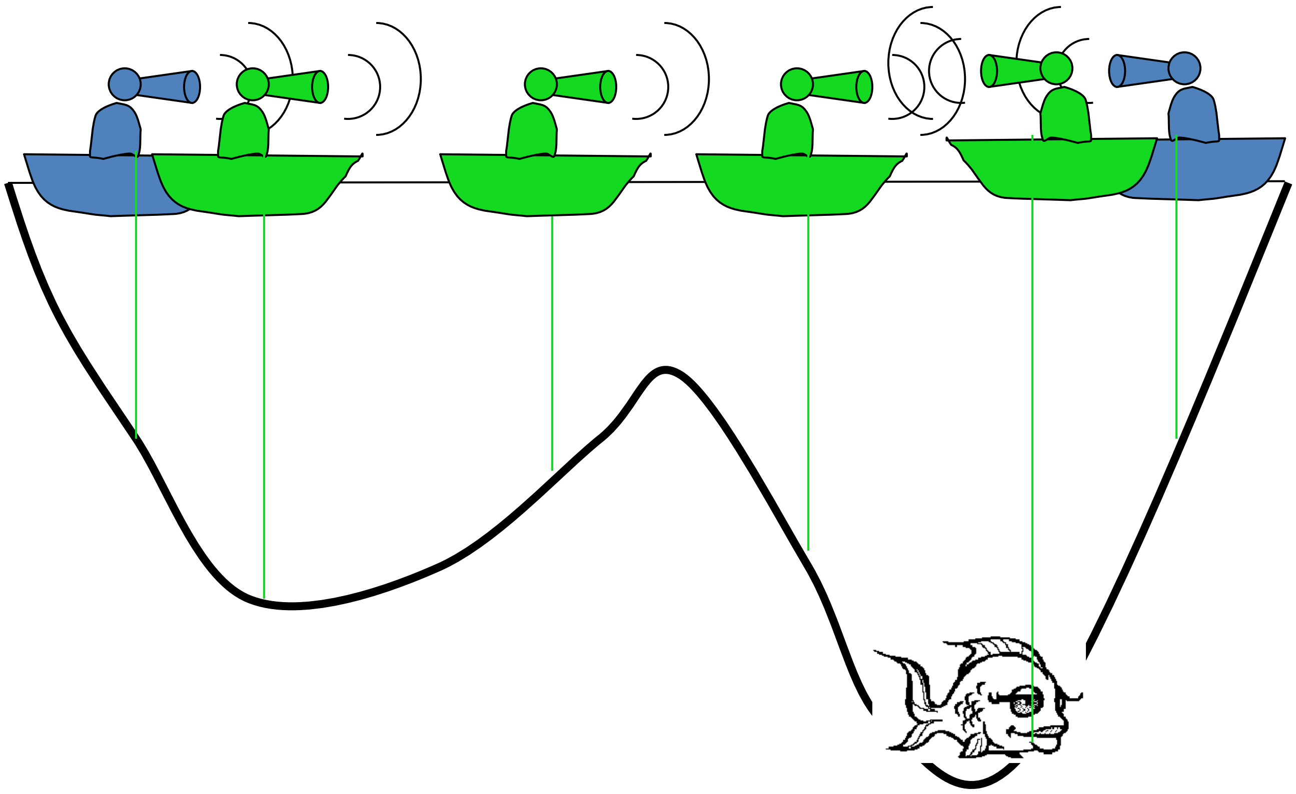
- **modelo de inteligencia colectiva**, que busca optimizar soluciones a un problema dado, utilizando un enjambre de partículas (población de soluciones), donde cada una se va optimizando, dependiendo de la experiencia personal, y de la experiencia de las otras partículas en su vecindario.

Estas experiencias, son denominadas por Engelbrecht, como **componente cognitivo y componente social**, respectivamente Partirle Swarm Optimization

Unión hace la fuerza



Para Cooperar



Generalidades

El algoritmo de enjambre de partículas se basa en una idea muy simple:

- la gente aprende a darle sentido al mundo, interactuando con los otros sobre él. Las personas hablan unas con otras, observan unas a otras, y aprenden unas de otras, para darle sentido a un mundo confuso.
- Esta premisa, permite diseñar algoritmos que codifican una población de individuos, que proponen soluciones a un problema dado, y luego son capaces de refinar las soluciones, mediante la interacción con sus "compañeras", recogiendo las sugerencias de sus vecinos, y a partir de ellas, se ajustan.
- Con el tiempo, los individuos encuentran buenas soluciones al problema, aun cuando el problema sea muy difícil.

En general, esta técnica ha sido utilizada en problemas multiobjetivos, dinámicos, y muchos otros tipos de problemas difíciles.

Características de las partículas

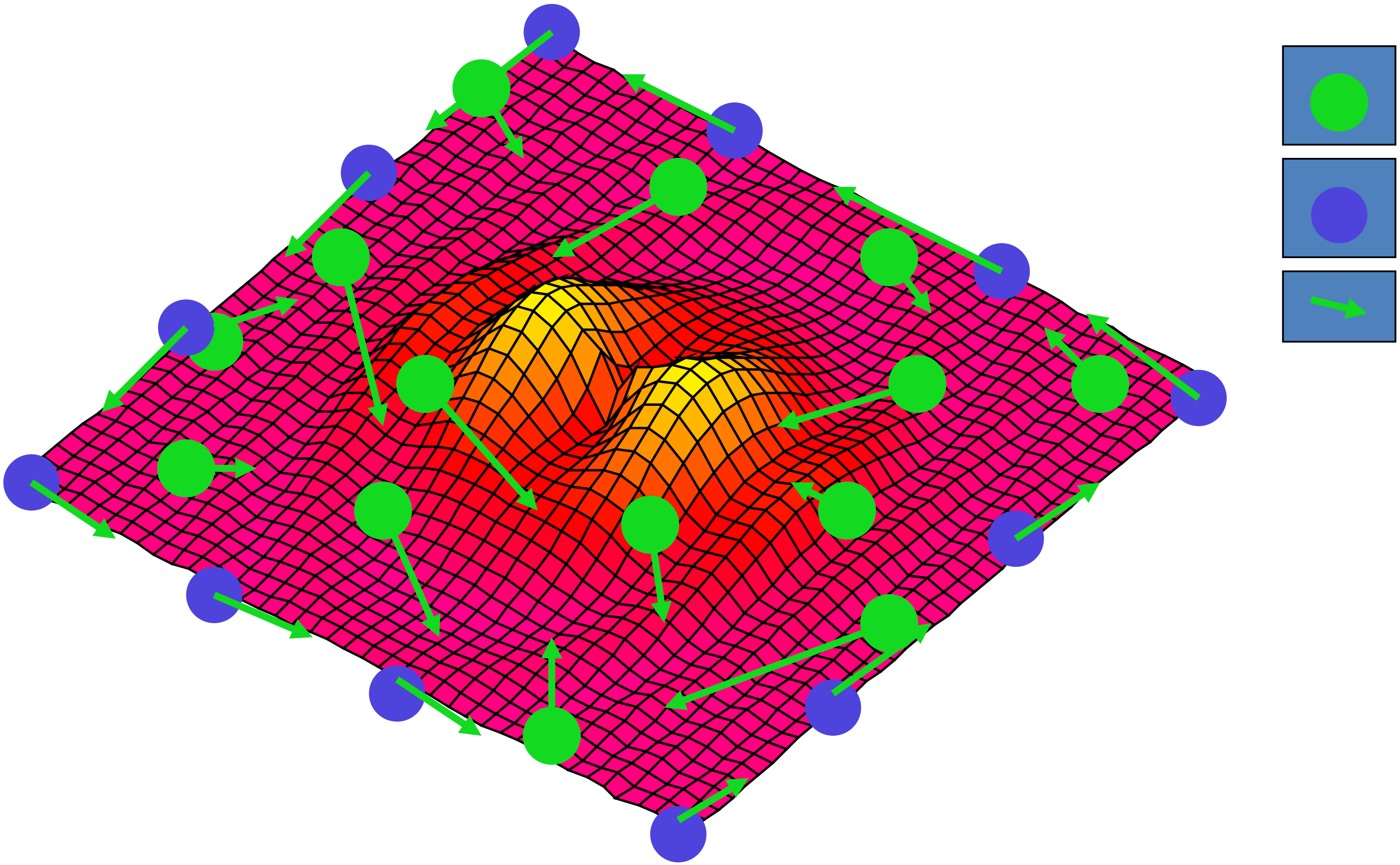
- Cada partícula posee :
  - **un vector de posición** (x_i), que es la solución codificada en un espacio de búsqueda en R^n, y
  - **un vector de velocidad** (v_i) también en R^n, que determina la magnitud del cambio por cada dimensión del vector de posición de la partícula.

Por lo tanto, a mayor velocidad, mayor es el cambio en la posición.

- **La posición de la partícula i** durante la iteración t+1, es definida por

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

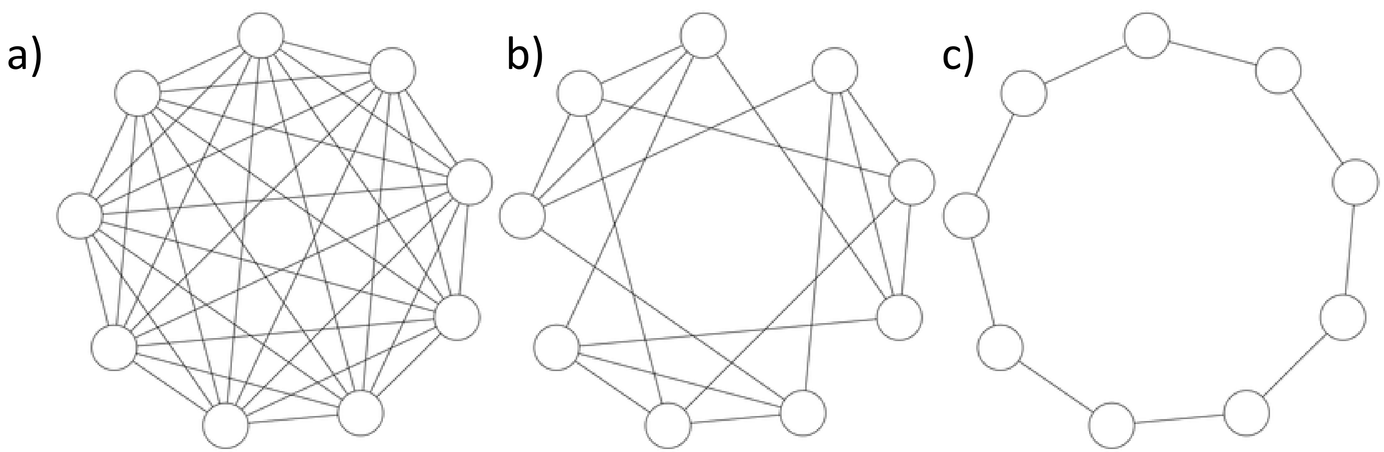
características de una partícula: posición y velocidad



Tipo de topología del enjambre

- **GBEST,** por su nombre en inglés (Global Best) que describe una población totalmente interconectada. En GBEST, aunque contiene el mayor número posible de conexiones entre los miembros de la población, en la práctica, significa seguir la mejor solución encontrada.
- **LBEST** (Local Best) es una red en anillo, donde cada partícula está conectada a las partículas a cada lado de ella. En esa estructura, las subpoblaciones pueden converger, de manera independiente, en diversos óptimos, en el espacio del problema.

Además, en la literatura se han **propuesto otras** (ver otras en http://www.scholarpedia.org/article/Particle_swarm_optimization)



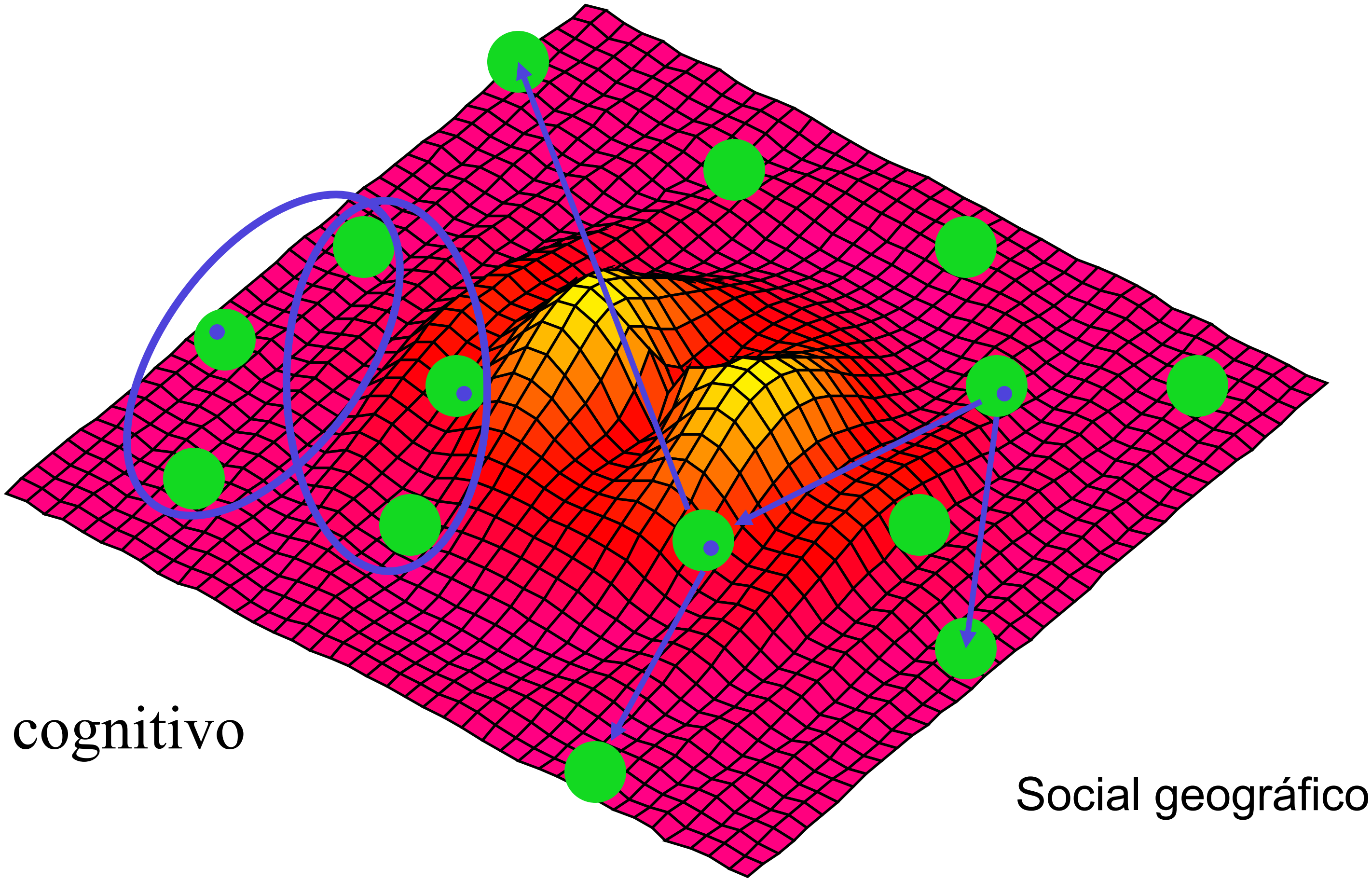
Otras características de los enjambres de partículas

- **Usan una norma de cambio** (normalmente, expresada en una fórmula matemática). Las partículas se mueven a través del espacio de búsqueda, tal que la selección de un punto en el tiempo t, es definida por una norma de cambio, que depende de su posición en el instante t-1, sus éxitos anteriores, y los éxitos anteriores de sus vecinos.

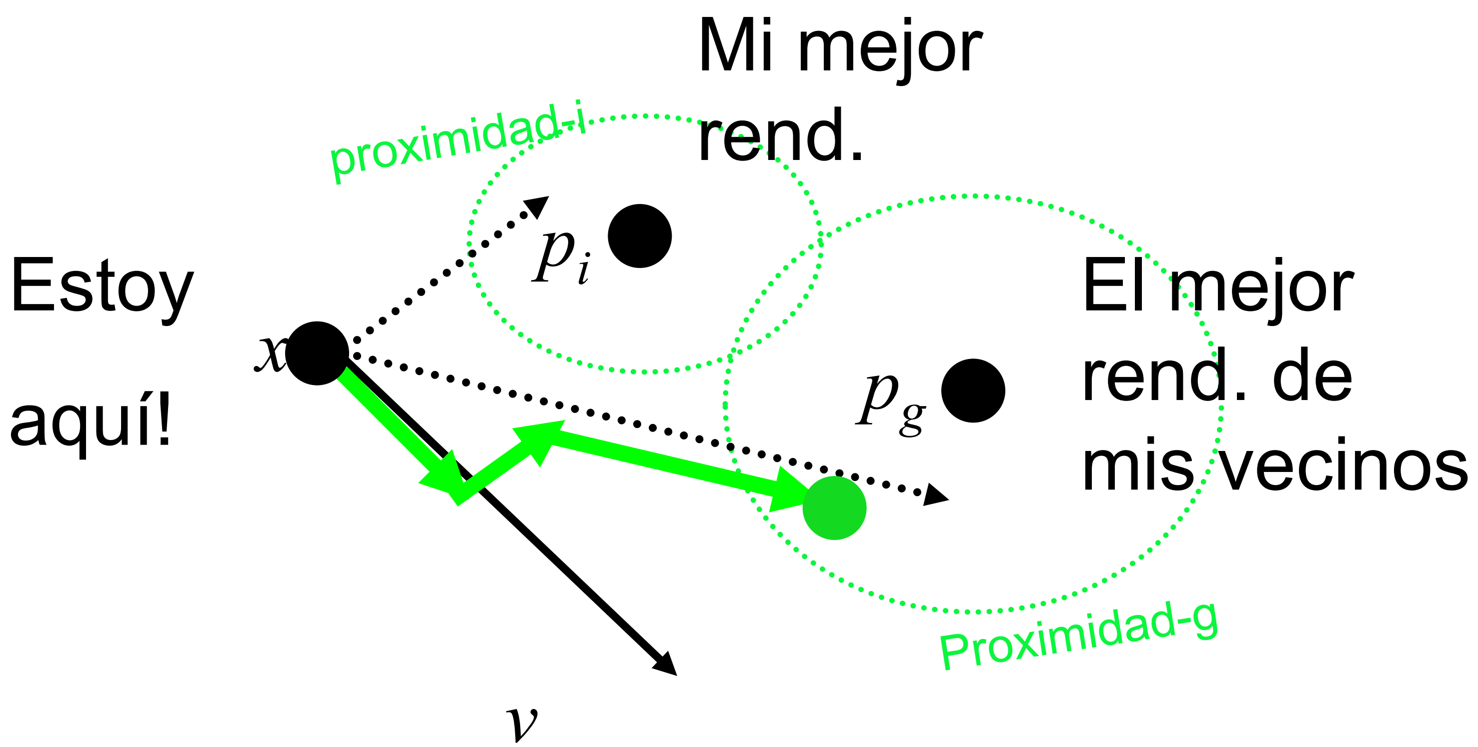
El *algoritmo de búsqueda, busca primero las regiones buenas en el medio ambiente (explora) y, a continuación, habiendo encontrado una buena región, busca el mejor punto en esa región (explota).*

- **La regla de interacción** consiste en: una partícula considera sus éxitos, y los de algunas otras partículas, en la determinación del siguiente punto de prueba, en el espacio de búsqueda.

Vecindad

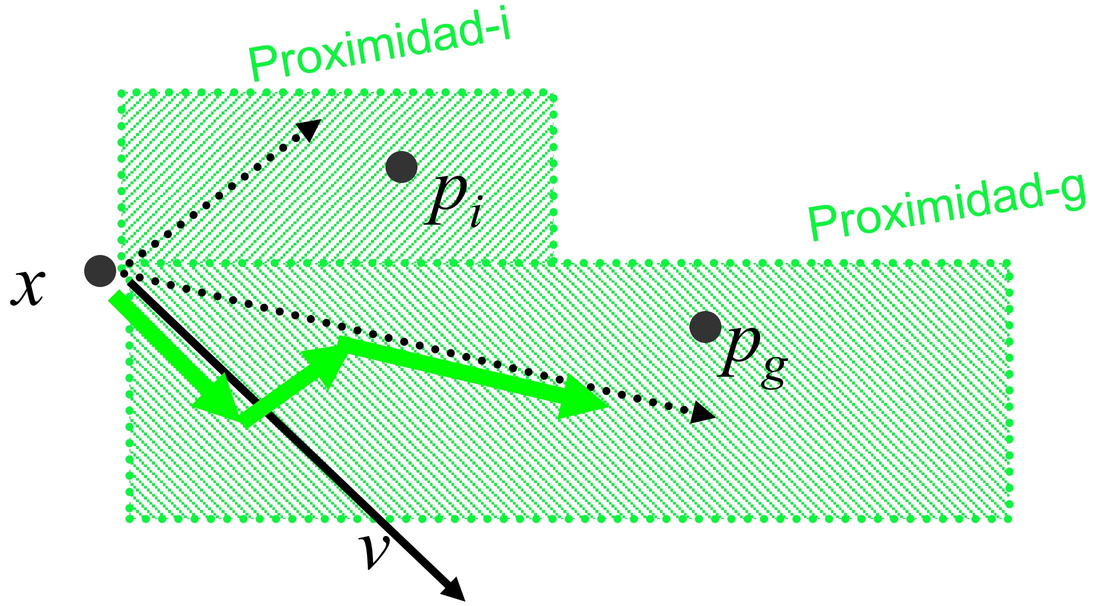


compromiso psicosocial



Aproximación aleatoria

Hiperparalelepipedo



Macroalgoritmo

En cada paso t

Para cada particula

Por cada componente d

La aleatoriedad
dentro del bucle

Actualizar la velocidad

$$
\begin{cases}
v_d(t+1) = \\
\alpha v_d(t) \\
+ \beta rand(0, \varphi_1)(p_{i,d} - x_d(t)) \\
+ \beta rand(0, \varphi_2)(p_{g,d} - x_d(t))
\end{cases}
$$

Despues moverse $x(t+1) = x(t) + v(t+1)$

Los parámetros

- **Tamaño del enjambre**: determina la cantidad de partículas que contiene el enjambre. Mientras mayor es esa cantidad, mejores son los resultados, porque se explora más el espacio de búsqueda. Sin embargo, el aumento del tamaño, también aumenta el tiempo computacional requerido para la hacer la evolución.

- **Tamaño del vecindario**: parámetro válido para los enfoques que usan vecindarios, como LBEST, y define la interacción social de las partículas en el enjambre. Mientras menor es el vecindario, menor es la interacción entre las partículas, pero aun así, se pueden encontrar buenas soluciones.

- **Número de iteraciones**: es dependiente del problema.

- **Coeficientes de aceleración**: generalmente se utilizan valores constantes. Estos valores, determinan la importancia del componente cognitivo (c_1) y del componente social (c_2).

  Engelbrecht considera que cuando $c_1<c_2$, la partícula es atraída con más fuerza hacia su mejor posición personal, en cambio, cuando $c_2<c_1$, la partícula es atraída con mayor fuerza hacia la mejor posición global;

Algoritmo GBEST

La actualización de la velocidad, en este caso, viene determinada por:

$$v_{ij}(t+1) = v_{ji}(t) + c_1 r_{1j}(t)\left[y_{ij}(t) - x_{ij}(t)\right] + c_2 r_{2j}(t)\left[\hat{y}_j(t) - x_{ij}(t)\right]$$

- Donde, $v_{ji}(t)$ es la velocidad de la partícula i en la dimensión j durante la iteración t, c_1 y c_2 son constantes de aceleración, que definen la importancia del componente cognitivo y social, respectivamente,
- r_{1j} y r_{2j} son números aleatorios en el rango [0, 1], sacados de distribuciones uniformes independientes,
- y_{ij} es la dimensión j de la mejor posición alcanzada por la partícula i (componente cognitivo), y
- $\hat{y}_j(t)$ es la dimensión j de la mejor posición global alcanzada por el enjambre hasta la iteración t (componente social).

Algoritmo GBEST

La mejor posición individual alcanzada por la partícula i, se calcula de acuerdo a

$$y_i(t+1) = \begin{cases} y_i(t) & si\ f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & si\ f\big(x_i(t+1)\big) < f(y_i(t)) \end{cases}$$

- Donde la función *f* define la proximidad de la partícula con el punto óptimo, tal que la mejor posición individual es un problema de minimización de *f*.

Algoritmo GBEST

La **mejor posición global alcanzada por el enjambre**, es la mejor entre las mejores posiciones individuales alcanzadas por las partículas que conforman el enjambre.

Matemáticamente es descrita por

$$\hat{y}(t) \in \{y_0(t), \ldots, y_{n_s}(t)\} | f(\hat{y}(t)) = \min \{f(y_o(t)), \ldots, f(y_n(t))\}$$

- Donde, n es el número total de partículas en el enjambre.

Algoritmo GBEST

- *Crear e inicializar un enjambre n-dimensional;*
- *Repetir hasta alcanzar condición de parada*
  - *por cada partícula* $i = 1, n$
    - *si* $f(x_i) < f(y_i)$ *entonces*
      - $y_i = x_i$
    - *si* $f(y_i) < f()$ *entonces*
      - $y_i = y_g$
  - *por cada partícula* $i = 1, n$
    - *actualizar la velocidad*
    - *actualizar la posición*

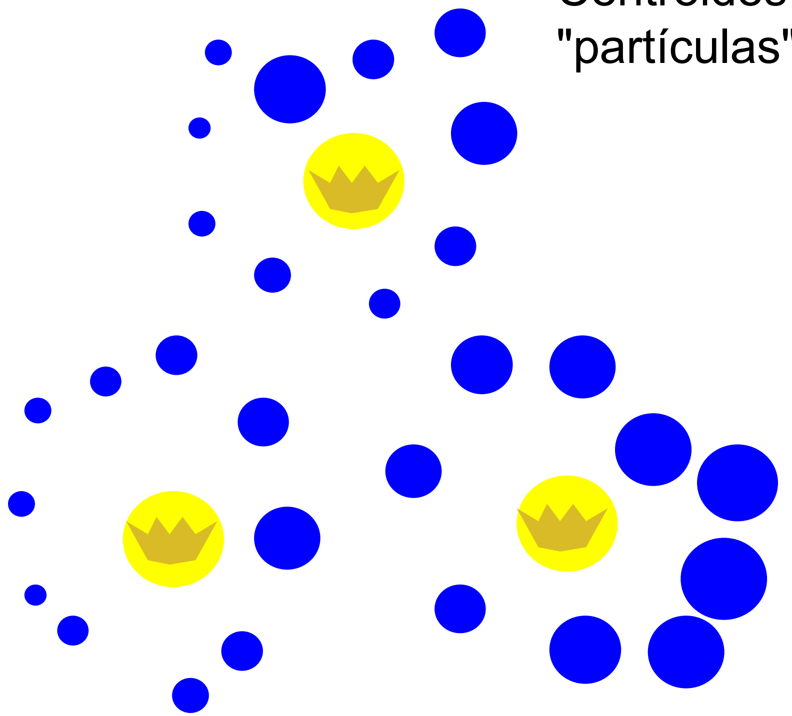
Algunas variantes

Grupos y reinas

Cada partícula es ponderado por su perf.

Centroides = reinas = "partículas" vecinas

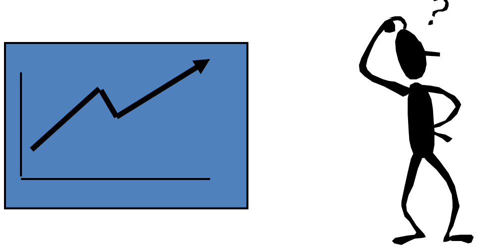
clustering dinámico



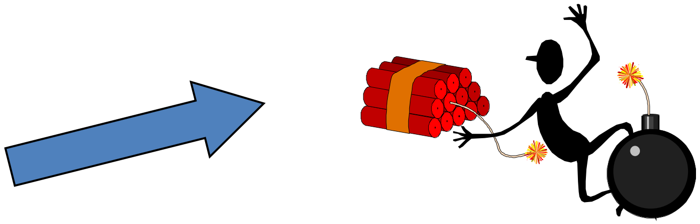
Tamaño enjambre Adaptativo

Ha habido una mejora suficiente

yo soy el peor

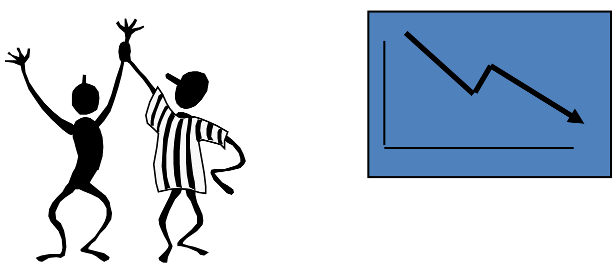


Yo trato de matarme

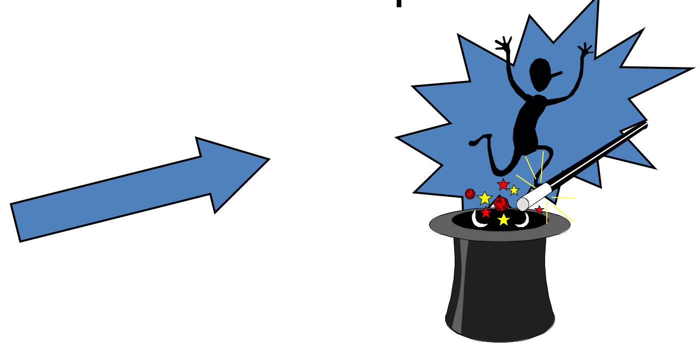


Yo soy el mejor

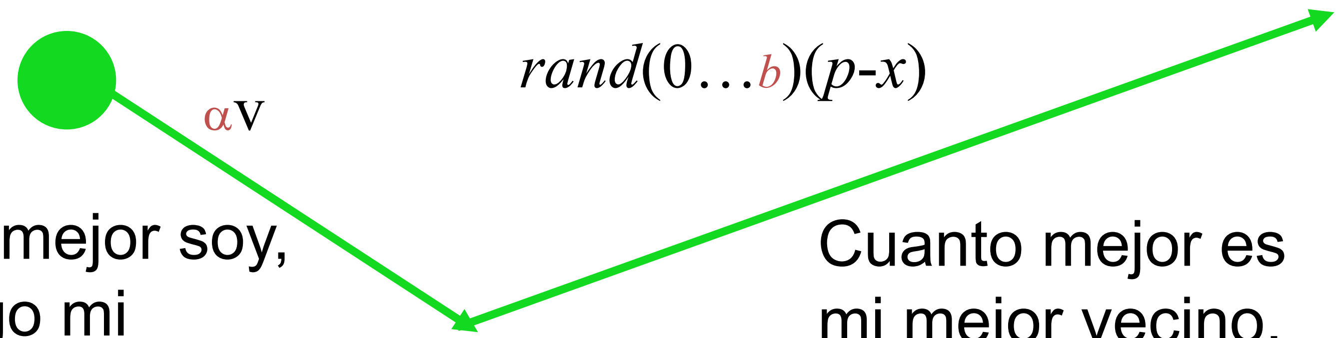
pero no ha sido suficiente la mejora



Trato de generar una nueva partícula



coeficientes adaptables



El mejor soy, sigo mi propio camino

Cuanto mejor es mi mejor vecino, más tiendo ir hacia él

Algunas mejoras

Límites de Velocidad: Los límites de velocidad se diseñan para controlar la exploración y refinar la búsqueda, evitando que las velocidades de las partículas alcancen valores tan altos, que los cambios en la posición sean tan grandes que puedan obviar áreas de interés en el espacio de búsqueda.

$$V_{max,j} = \delta(x_{max,j} - x_{min,j})$$

- donde $x_{max,j}$ y $x_{minx,j}$ son los valores máximos y mínimos de x en la dimensión j,
- $\delta \in [0, 1]$

Algunas mejoras

Una vez establecido el límite de velocidad por cada dimensión, se determina cómo **hacer cumplir el límite de velocidad.**

$$v_{ij}(t+1) = \begin{cases} v'_{ij}(t+1) & si\ v'_{ij}(t+1) < V_{max,j} \\ V_{max,j} & si\ v'_{ij}(t+1) \geq V_{max,j} \end{cases}$$

o

$$v_{ij}(t) = V_{max,j} \tanh\left(\frac{v'_{ij}(t+1)}{V_{max,j}}\right)$$

- La importancia de $V_{max,j}$ radica en que, para un valor pequeño de $V_{max,j}$ se favorece la explotación, porque los cambios de posición son pequeños, mientras que para un valor grande se favorece la exploración

Algunas mejoras

- ***Peso de Inercia:*** El peso de inercia de una partícula determina la influencia de la velocidad anterior $v_{ij}(t)$, al momento de calcular la nueva velocidad $v_{ij}(t+1)$.
  - Se trata de otro mecanismo, para controlar entre explorar y explotar en el enjambre.
  - Sustituye la ecuación de actualización de velocidad del algoritmo GBEST (para LBEST se realiza un cambio similar), donde w es el peso de inercia.

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}(t)\left[y_{ij}(t) - x_{ij}(t)\right] + c_2 r_{2j}(t)\left[\hat{y}_j(t) - x_{ij}(t)\right]$$

Algunas mejoras

- Según Engelbrecht, el **valor de w es muy importante**, porque de él depende la convergencia del enjambre y el equilibrio óptimo entre explorar y explotar.

- Engelbrecht propone diferentes enfoques de **ajuste dinámico del peso de inercia:** a) ajustes aleatorios, b) disminución lineal, c) disminución no-lineal, d) inercia difusa adaptativa, y e) aumento lineal.

- Por ejemplo, en la **disminución lineal** se va disminuyendo linealmente el peso de inercia:

$$w(t) = \left(w(0) - w(n_t)\right)\frac{(n_t - t)}{t} + w(n_t)$$

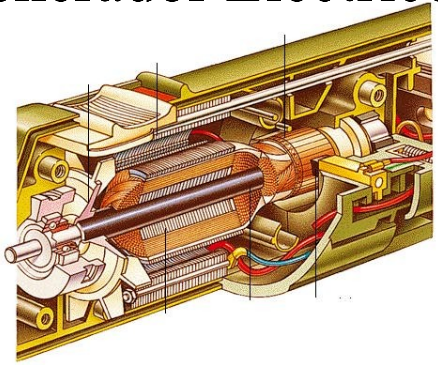
- Donde n_t es el número máximo de iteraciones en que el algoritmo será ejecutado,
- w(0) es el peso inicial de inercia,
- $w(n_t)$ es el peso final de inercia, y
- w(t) es el peso de inercia en la iteración t (note que w(0) debe ser mayor a $w(n_t)$).

Aplicaciones

Medicas

Industriales

Generador Electrico



Vehículos Electricos

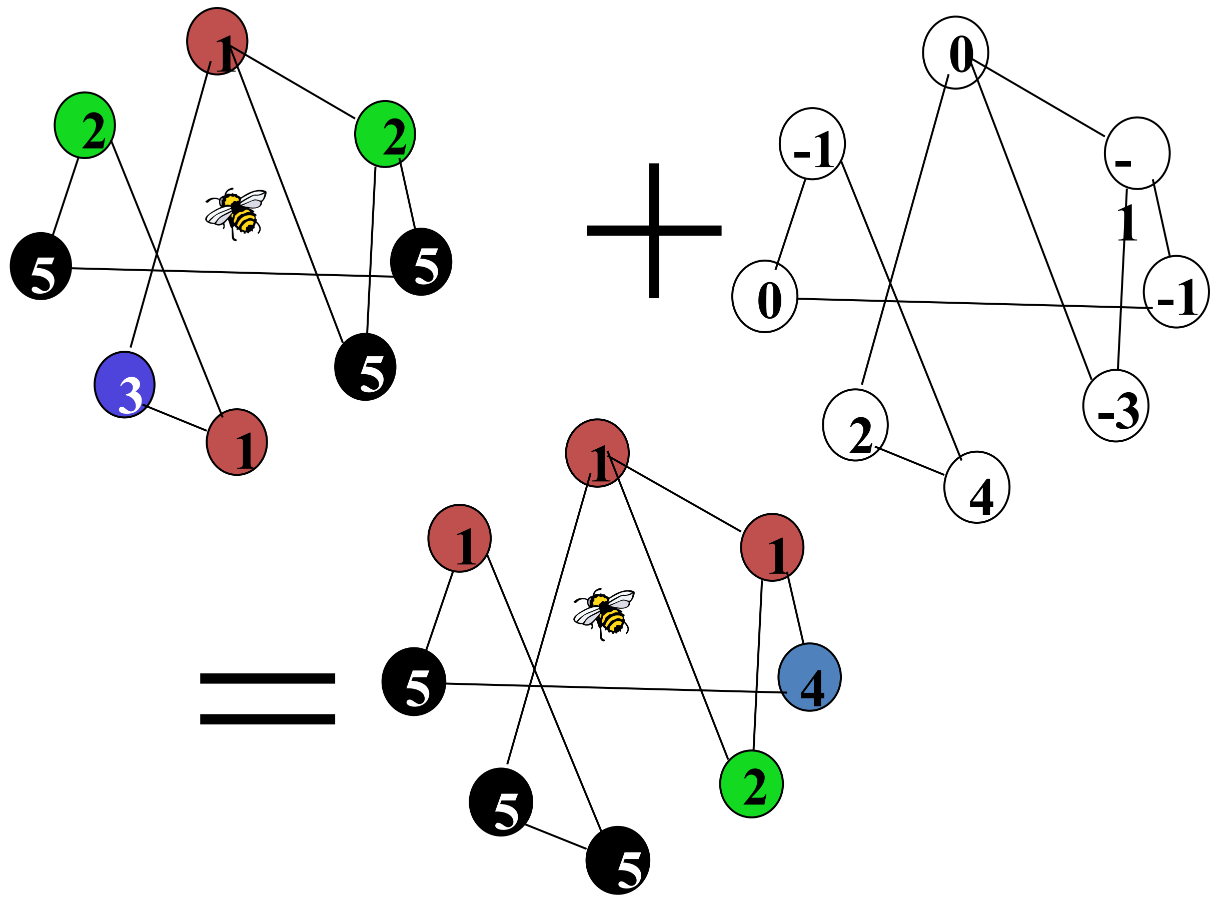
Problema de la Mochila

granularidad=1

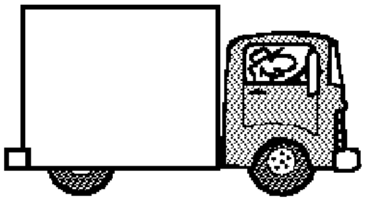
$$\begin{cases} x_i \in \{1 \ldots N\} \\ i \neq j \Rightarrow x_i \neq x_j \\ \displaystyle\sum_{i \in I, |I| = D, I \subset \{1, N\}} x_i = S \end{cases}$$

N=100, D=10, S=100,
870 evaluaciones:
corrida1 => (9, 14, 18, 1, 16, 5, 6, 2, 12, 17)
corrida 2 => (29, 3, 16, 4, 1, 2, 6, 8, 26, 5)

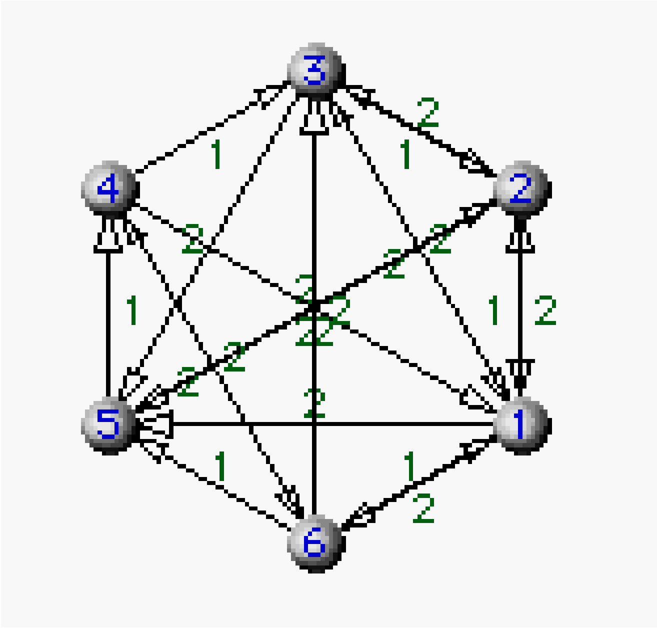
Problema Grafo Coloreado

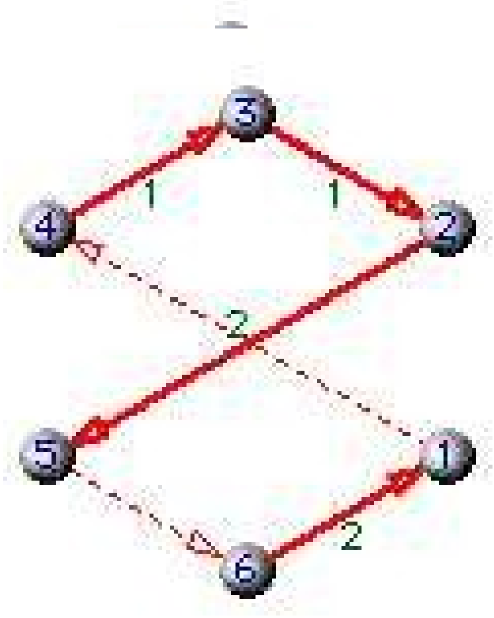


Viajero de comercio



Ejemplo de posición: X=(5,3,4,1,2,6)
Ejemplo de velocidad: v=((5,3),(2,5),(3,1))





Aplicaciones

Cockshott A. R., Hartman B. E., "Improving the fermentation medium for Echinocandin B production. Part II: Particle swarm optimization", *Process biochemistry*, vol. 36, 2001, p. 661-669.

He Z., Wei C., Yang L., Gao X., Yao S., Eberhart R. C., Shi Y., "Extracting Rules from Fuzzy Neural Network by Particle Swarm Optimization", *IEEE International Conference on Evolutionary Computation*, Anchorage, Alaska, USA, 1998.

Secrest B. R., Traveling Salesman Problem for Surveillance Mission using Particle Swarm Optimization, AFIT/GCE/ENG/01M-03, Air Force Institute of Technology, 2001.

Yoshida H., Kawata K., Fukuyama Y., "A Particle Swarm Optimization for Reactive Power and Voltage Control considering Voltage Security Assessment", *IEEE Trans. on Power Systems*, vol. 15, 2001, p. 1232-1239.

Magic Square

55 30 68
42 49 62
56 74 23

30 61 53
89 32 23
25 51 68

43 51 78
75 33 64
54 88 30

80 3 30
22 72 19
11 38 64

50 43 67
58 55 47
52 62 4

65 28 64
63 55 39
29 74 54

D=3x3, N=100
10 corridas
13430 evaluaciones

10 soluciones

18 25 59
32 53 17
52 24 26

50 65 68
69 42 72
64 76 43

27 96 39
73 40 49
62 26 74

22 70 58
40 75 35
88 5 57

Otros modelos de optimización por enjambre de partículas,

- Algoritmo de Búsqueda Gravitacional (Gravitational search algorithm)
- Modelo de Gotas de Agua Inteligentes (Intelligent Water Drops)
- Modelo de Formación Dinámica Fluvial (River formation dynamics)
- Algoritmo de Búsqueda por Difusión Estocástica (Stochastic diffusion search)

Algoritmo de Búsqueda Gravitacional (GSA)

- fue propuesto por Rashedi et al. en 2009, inspirado en la **ley de la gravitación universal**
- las masas se atraen entre sí por campo gravitatorio producido por sí mismos.
- La **fuerza gravitacional** entre dos partículas es directamente proporcional al producto de sus masas e inversamente proporcional al cuadrado de la distancia entre ellas

$$F = G\frac{M_1 M_2}{R^2}$$

G es una constante gravitacional

Algoritmo de Búsqueda Gravitacional

- De acuerdo a la segunda ley de Newton, la **aceleración de una partícula** se escribe como

$$a = \frac{F}{M}$$

- En GSA, inspirado en la ley de la gravitación universal de Newton y segunda ley de Newton de movimiento, las partículas o agentes actualicen sus lugares en una forma similar.

buenas soluciones significan masas pesadas. En otras palabras, cuanto más pesado es la masa, fuerza mayor aplicará a los demás, y una aceleración menor se obtendrá

Algoritmo de Búsqueda Gravitacional

En GSA cada agente tiene cuatro propiedades:

- Su posición, representa una solución del problema
- Su masa inercial M_i : medida de capacidad de resistencia para un agente para cambiar su estado de movimiento cuando se le aplica una fuerza
- Su masa activa M_a : medida de la fuerza del campo gravitatorio producido por el propio agente
- Su masa pasiva M_p : medida de la fuerza de la interacción con el mundial campo gravitatorio.

Las masas gravitacionales e inerciales son determinados por la función de aptitud

Algoritmo de Búsqueda Gravitacional

- sistema artificial con N agentes (masas), la posición del agente de i-ésimo es

$$X_i = (x_i^1, x_i^2, \ldots, x_i^D) \qquad \text{para i=1, N}$$

Donde x_i^d significa la posición del agente i-ésimo en la dimensión d, mientras que D es el dimensión global del espacio de búsqueda

- En el tiempo t, la aplicación de fuerza sobre la masa i desde j masa es

$$F_{ij}^d = G(t)\frac{M_{pi}(t)M_{ai}(t)}{R_{ij}(t)+\varepsilon}(x_j^d(t) - x_i^d(t))$$

ε es una constante para garantizar que el denominador no es cero.
Rij t) significa distancia euclidiana entre los agentes i y j.

Algoritmo de Búsqueda Gravitacional

La fuerza resultante que actúa sobre el agente i en la dimensión d es una suma ponderada al azar del componente d de la fuerza de los agentes Kbest

$$F_i^d(t) = \sum_{j \in Kbest, j \neq i} rand_j F_{ij}^d(t)$$

donde j rand es un número aleatorio entre 0 an 1. Kbest significa el conjunto de los primeros K agentes con mejor valor de fitness y mayor masa

Algoritmo de Búsqueda Gravitacional

De acuerdo con la ley del movimiento, la aceleración del agente i en el momento t en la dimensión d se expresa como

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)}$$

La velocidad y posición siguiente del agente se calcula como

$$v_i^d(t) = rand_i v_i^d(t) + a_i^d(t)$$

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

La constante gravitacional G se inicializa al comienzo y se reducirá

$$G(t) = G_o e^{-\frac{1}{T}}$$

donde Go es una constante inicial y T es el número total de iteración

Algoritmo de Búsqueda Gravitacional

Se calculan los valores de las masas $M_{ai} = M_{pi} = M_{ii} = M_i$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}$$

Donde $fit_i(t)$ indica el valor de la aptitud del agente i en el momento t.

Algoritmo GSA

1. Iniciar la población (generar una población aleatoria de N soluciones N)
2. Evaluar el valor de aptitud para cada agente
3. Actualización de G(t), best(t), worst(t), y $M_i(t)$ para i = 1,2, ..., N.
4. Calcular la fuerza total en direcciones diferentes para cada agente
5. Calcular la aceleración y la velocidad para cada agente.
6. Actualizar la posición de cada agente.
   Repita los pasos 2 a 6 hasta que se cumpla el criterio de parada.

Conclusión GSA

1. Una masa de agente se evalúa por su valor de fitness. Cada agente puede observar el rendimiento de los otros y se ve afectado por otros agentes alrededor de sí mismo por fuerza gravitacional aplicado en él, en lugar de seguir el mejor agente.
2. Un agente con una buena fitness significa una masa pesada, una gran intensidad de atracción hacia los demás y un movimiento lento.
3. GSA es algoritmo sin memoria. Pero sigue siendo eficaz como los algoritmos con memoria. La Constante gravitacional disminuye con el tiempo y se ajusta la precisión de búsqueda, similar al peso de la inercia adaptativa en PSO