Funcionamiento algoritmo PSO

Joaquín F Sánchez

Marzo - 2017

Curso de computación Adapativa

Agenda

- Funcionamiento Algoritmo
- Simulación en Matlab
- Resultados gráficos

Parámetros del PSO

Vector velocidad

$$V_i = (v_{i1}, v_{i2}, ..., v_{iN}) \tag{1}$$

Variable de estado inicializada aleatoriamente que identifica a una partícula i.

Vector posición

$$X_i = (x_{i1}, x_{i2}, ..., x_{iN}) \tag{2}$$

Variable de estado inicializada aleatoriamente que identifica a una partícula i.

Corresponde a una solución potencial de optimización.

Espacio de búsqueda

$$x_n \in [x_{n,min}, x_{n,max}] \tag{3}$$

Conjunto de límites de las variables a optimizar en el cual debe restringirse el movimiento de las partículas.

Parámetros del PSO

Memoria o Nostalgia

$$P_i = (p_{i1}, p_{i2}, ..., p_{in}) \tag{4}$$

Registro de la posición espacial asociada con la mejor solución históricamente visitada por cada partícula.

Mejor Posición Histórica

$$G = (g_1, g_2, ..., g_N) \tag{5}$$

Término usado para designar la mejor posición histórica encontrada por todos los congéneres comparada con todas las generaciones anteriores.

Movimiento de las partículas

En cada iteración del método, k, cada una de las partículas de la población recorre el espacio de soluciones con una velocidad V_i hacia nuevas posiciones X_i de acuerdo con su propia experiencia P_i y con la experiencia aportada por el mejor de sus congéneres, G

$$v_{in}(k+1) = \omega v_{in}(k) + c_i r_1(k) \cdot (p_{in}(k) - x_{in}(k)) + c_2 r_2(k) \cdot (g(k) - x_{in}(k)) \quad (6)$$

$$x_{in}(k+1) = x_{in}(k) + \tau v_{in}(k+1) \quad (7)$$

$\omega, c_1, c_2, \tau \geq 0$

- ω: coeficiente de inercia de la partícula.
- c_1: constante de aceleración cognitiva.
- c_2: constante de aceleración social.
- τ: factor de construcción

Movimiento de las partículas

- $v_{in}(k+1)$: momento, habito
- v_{in}: inercia
- $c_1 r_1(k) \cdot (p_{in}(k) - x_{in}(k))$: memoria, nostalgia o auto aprendizaje.
- $c_2 r_2(k) \cdot (g(k) - x_{in}(k))$: Cooperación, conocimiento de grupo
- $[-v_{max}, v_{max}]$: intervalo de restricción de la velocidad.

El algoritmo PSO Simple

1. Iniciar un arreglo de partículas con velocidades y posiciones aleatorias dentro del espacio de búsqueda.
2. Evaluar la función objetivo para cada partícula.
3. Comparar el valor de la función objetivo actual con el valor de P_i, poner el valor actual en P_i
4. Identificar la partícula del enjambre con el mejor valor de la función objetivo encontrado hasta el momento y poner las coordenadas de está posición en G.
5. Ajustar la velocidad V_i y la posición X_i de acuerdo con las ecuaciones.
6. Si se alcanza el criterio de terminación, parar el algoritmo; en caso contrario volver al paso 2.

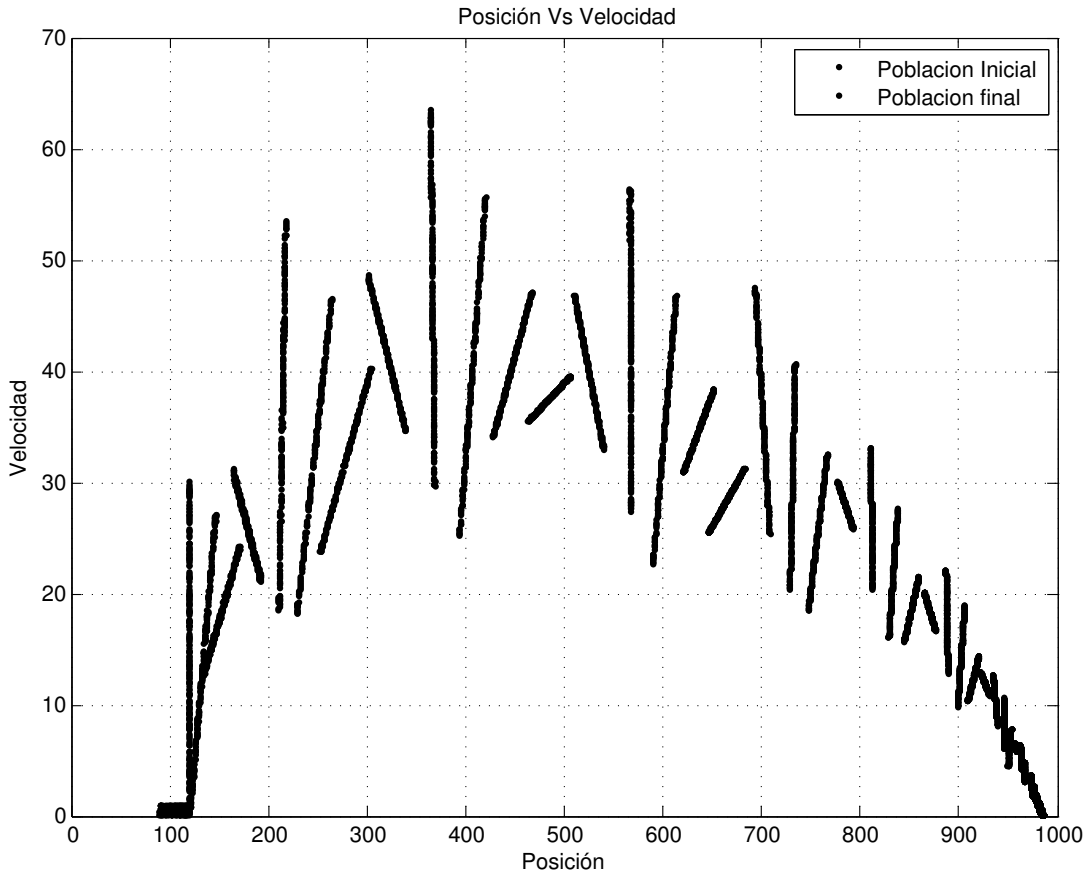
Código en MATLAB

```
clear all
clc
N=250;% Numero de particulas
maxit=100;% Numero de pasadas
dim =1;
upbnd = 90; % Limite superior de inicio del swarm
lwbnd = 120; % Limite inferior de inicio del swarm
% Inicializa velocidad y posición del swarm.
x = rand(N,dim)*(upbnd-lwbnd) + lwbnd ;
v = rand(N,dim); %velocidad inicial
plot(x,v,'.r')
hold on
title('Posición Vs Velocidad')
grid on
xlabel('Posición')
ylabel('Velocidad')
[brs,kol]=size(x);
f=zeros(N,1);
rhomax=0.9;rhomin=0.4;
for it=1:maxit
    rho(it)=rhomax-((rhomax-rhomin)/maxit)*it;
end
```

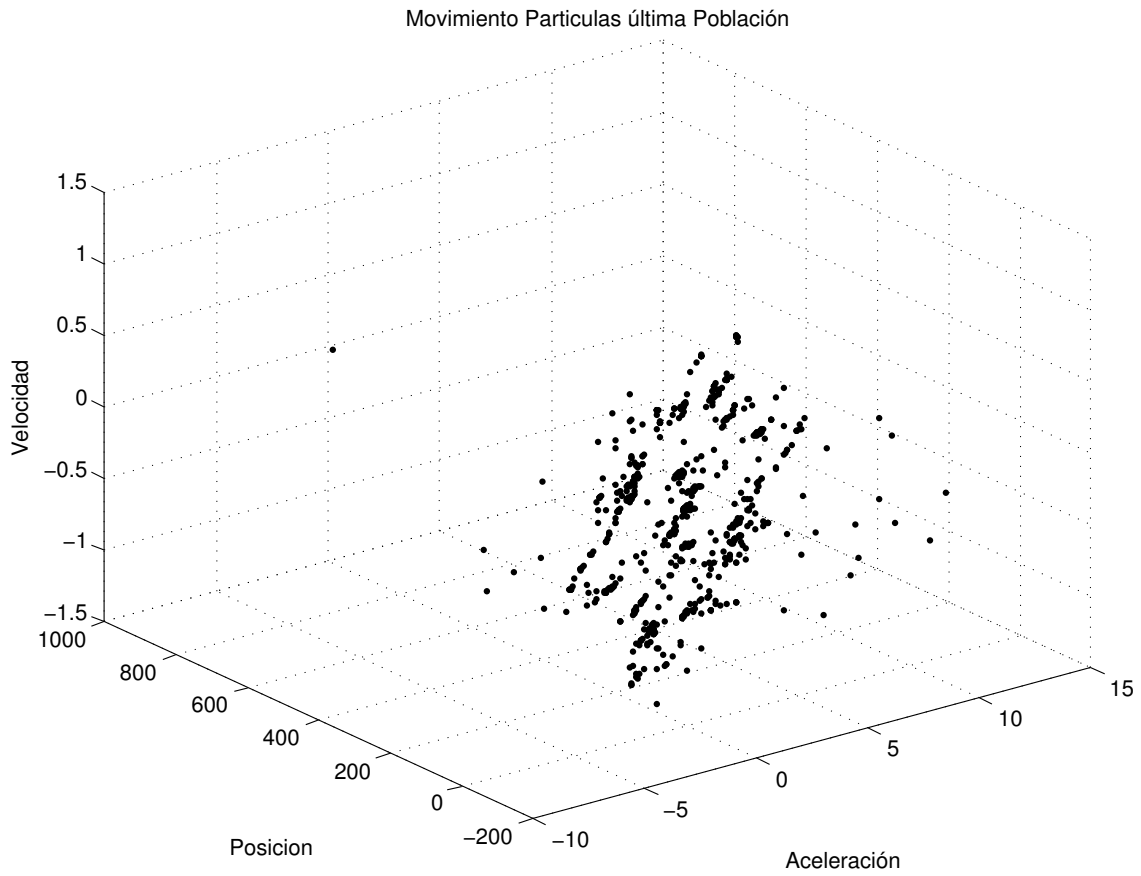
```
%Evaluación de la función objetivo en cada particula
for i=1:brs
    f(i)=fungsi2(x(i,:));
end
%Inicia el proceso de comparación
it=1;
Pbest=x;
fbest=f;
[minf,idk]=min(f);
Gbest=x(idk,:);
lastbest=[0 0];
minftot=[];
while it<maxit
    r1=rand;r2=rand;
    for j=1:brs
        v(j,:)=rho(it).*v(j,:)+r1.*(Pbest(j,:)-x(j,:))+r2.*(Gbest-x(j,:));
        x(j,:)=x(j,:)+v(j,:);
        f(j)=fungsi2(x(j,:)); % Se evalua la función objetivo
     end
    %Se actualiza Pbest
    changerow = f < fbest;
    fbest=fbest.*(1-changerow)+f.*changerow;
    Pbest(find(changerow),:)=x(find(changerow),:);
    [minf,idk]=min(fbest);
    minftot=[minftot;minf];
    Gbest=Pbest(idk,:);
    if sum(var(Pbest))<1e-8
        break
    end
```

```
 it=it+1;
    lastbest=Gbest;
    figure(1)
    plot(x,v,'.');
    pause(0.01)
    legend('Poblacion Inicial','Poblacion final')
end
x(1,3)=0;
v(1,3)=0;
pos=x;
vel=v;
grafica(N,it,v(length(v)),pos,vel)
xopt=Gbest;
fmin=minf;
figure(3)
plot(minftot,'r*')
grid on
title('Valores optimos')
```

Resultado del movimiento de las partículas



Resulatdo en 3D de la ultima población



Resultado de los valores optimo