

# Galactic Swarm Optimization

A new global optimization metaheuristic inspired by galactic motion

Francisco Solano López Rodríguez

- 1 Introducción
- 2 Particle swarm optimization
- 3 Galaxy swarm optimization
- 4 Algoritmo
- 5 Código

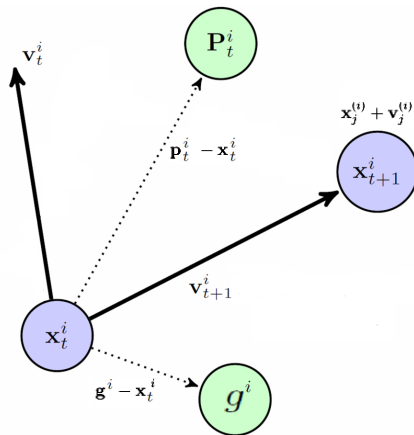
# Introducción

La metaheurística GSO se inspira en el movimiento que describen las estrellas en el interior de una galaxia, y además a una mayor escala, en el movimiento de las galaxias pertenecientes a un cúmulo. A su vez un cúmulo podría pertenecer a un supercúmulo y de esta forma podríamos seguir aumentando la escala.



- 1 Introducción
- 2 Particle swarm optimization**
- 3 Galaxy swarm optimization
- 4 Algoritmo
- 5 Código

# Particle swarm optimization



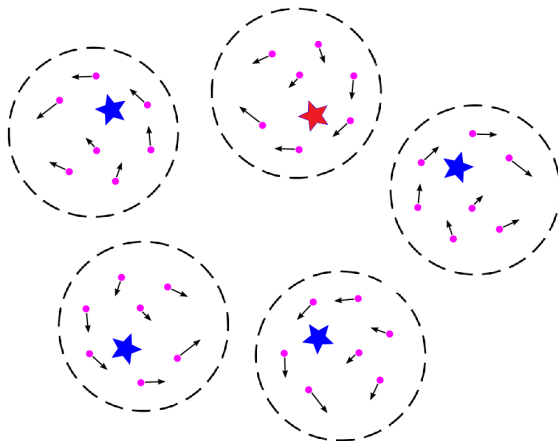
$$\mathbf{v}_j^{(i)} \leftarrow w_1 \mathbf{v}_j^{(i)} + c_1 r_1 (\mathbf{p}_j^{(i)} - \mathbf{x}_j^{(i)}) + c_2 r_2 (\mathbf{g}^{(i)} - \mathbf{x}_j^{(i)})$$

$$\mathbf{x}_j^{(i)} \leftarrow \mathbf{x}_j^{(i)} + \mathbf{v}_j^{(i)}$$

- 1 Introducción
- 2 Particle swarm optimization
- 3 Galaxy swarm optimization**
- 4 Algoritmo
- 5 Código

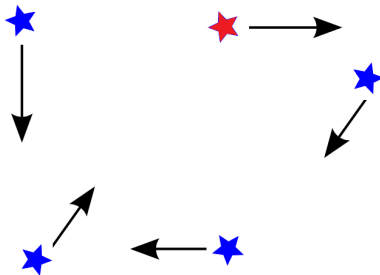
- Hay 2 niveles, los cuales se alternaran. El **nivel 1** será el que realice la **exploración**, mientras que el **nivel 2** realizará la **explotación**.
- **Nivel 1**: movimiento de las estrellas dentro de cada galaxia usando el algoritmo PSO, donde cada estrella tiene una velocidad, un mejor valor personal y el mejor global que persiguen se corresponde con la mejor solución hallada de su galaxía.
- **Nivel 2**: movimiento de las galaxías. Se inicializan soluciones con las mejores soluciones de cada galaxia. Cada una de las soluciones se mueve como un PSO. Ahora el mejor global que se persigue es la mejor solución encontrada hasta el momento.

# Galaxy swarm optimization Nivel 1

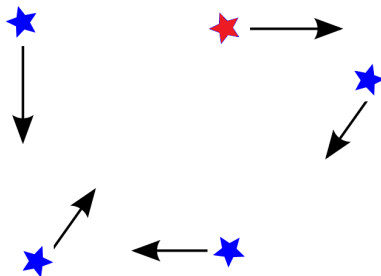




# Galaxy swarm optimization Nivel 2



## Galaxy swarm optimization Nivel 2



$$v^{(i)} \leftarrow w_2 v^{(i)} + c_3 r_3 (p^{(i)} - y^{(i)}) + c_4 r_4 (g - y^{(i)})$$
$$x_j^{(i)} \leftarrow x_j^{(i)} + v_j^{(i)}$$

- 1 Introducción
- 2 Particle swarm optimization
- 3 Galaxy swarm optimization
- 4 **Algoritmo**
- 5 Código

# Algoritmo

Level 1 Initialization:  $\mathbf{x}_j^{(i)}, \mathbf{v}_j^{(i)}, \mathbf{p}_j^{(i)}, \mathbf{g}^{(i)}$  within  $[x_{min}, x_{max}]^D$  randomly.

Level 2 Initialization:  $\mathbf{v}^{(i)}, \mathbf{p}^{(i)}, \mathbf{g}$  within  $[x_{min}, x_{max}]^D$  randomly.

for  $EP \leftarrow 1$  to  $EP_{max}$

```

do {
    Begin PSO: Level 1
    for  $i \leftarrow 1$  to  $M$ 
        do {
            for  $k \leftarrow 0$  to  $L_1$ 
                do {
                    for  $j \leftarrow 1$  to  $N$ 
                        do {
                             $\mathbf{v}_j^{(i)} \leftarrow \omega_1 \mathbf{v}_j^{(i)} + c_1 r_1 (\mathbf{p}_j^{(i)} - \mathbf{x}_j^{(i)}) + c_2 r_2 (\mathbf{g}^{(i)} - \mathbf{x}_j^{(i)});$ 
                             $\mathbf{x}_j^{(i)} \leftarrow \mathbf{x}_j^{(i)} + \mathbf{v}_j^{(i)};$ 
                            if  $f(\mathbf{x}_j^{(i)}) < f(\mathbf{p}_j^{(i)})$ 
                                do {
                                     $\mathbf{p}_j^{(i)} \leftarrow \mathbf{x}_j^{(i)};$ 
                                    if  $f(\mathbf{p}_j^{(i)}) < f(\mathbf{g}^{(i)})$ 
                                        then {
                                             $\mathbf{g}^{(i)} \leftarrow \mathbf{p}_j^{(i)};$ 
                                            if  $f(\mathbf{g}^{(i)}) < f(\mathbf{g})$ 
                                                then  $\mathbf{g} \leftarrow \mathbf{g}^{(i)};$ 
                                        }
                                }
                        }
                    }
                }
            }
        }
    }
    Begin PSO: Level 2
    Initialize Swarm  $\mathbf{y}^{(i)} = \mathbf{g}^{(i)} : i = 1, 2, \dots, M;$ 
    for  $k \leftarrow 0$  to  $L_2$ 
        do {
            for  $i \leftarrow 1$  to  $M$ 
                do {
                     $\mathbf{v}^{(i)} \leftarrow \omega_2 \mathbf{v}^{(i)} + c_3 r_3 (\mathbf{p}^{(i)} - \mathbf{y}^{(i)}) + c_4 r_4 (\mathbf{g} - \mathbf{y}^{(i)});$ 
                     $\mathbf{y}^{(i)} \leftarrow \mathbf{y}^{(i)} + \mathbf{v}^{(i)};$ 
                    if  $f(\mathbf{y}^{(i)}) < f(\mathbf{p}^{(i)})$ 
                        do {
                             $\mathbf{p}^{(i)} \leftarrow \mathbf{y}^{(i)};$ 
                            if  $f(\mathbf{p}^{(i)}) < f(\mathbf{g})$ 
                                then  $\mathbf{g} \leftarrow \mathbf{p}^{(i)};$ 
                        }
                }
            }
        }
    }
    Return  $\mathbf{g}, f(\mathbf{g})$ 

```

# Algoritmo

Level 1 Initialization:  $\mathbf{x}_j^{(i)}, \mathbf{v}_j^{(i)}, \mathbf{p}_j^{(i)}, \mathbf{g}^{(i)}$  within  $[x_{min}, x_{max}]^D$  randomly.

Level 2 Initialization:  $\mathbf{v}^{(i)}, \mathbf{p}^{(i)}, \mathbf{g}$  within  $[x_{min}, x_{max}]^D$  randomly.

for  $EP \leftarrow 1$  to  $EP_{max}$

```

do {
    Begin PSO: Level 1
    for  $i \leftarrow 1$  to  $M$ 
        do {
            for  $k \leftarrow 0$  to  $L_1$ 
                do {
                    for  $j \leftarrow 1$  to  $N$ 
                        do {
                             $\mathbf{v}_j^{(i)} \leftarrow \omega_1 \mathbf{v}_j^{(i)} + c_1 r_1 (\mathbf{p}_j^{(i)} - \mathbf{x}_j^{(i)}) + c_2 r_2 (\mathbf{g}^{(i)} - \mathbf{x}_j^{(i)})$ ;
                             $\mathbf{x}_j^{(i)} \leftarrow \mathbf{x}_j^{(i)} + \mathbf{v}_j^{(i)}$ ;
                            if  $f(\mathbf{x}_j^{(i)}) < f(\mathbf{p}_j^{(i)})$ 
                                do {
                                     $\mathbf{p}_j^{(i)} \leftarrow \mathbf{x}_j^{(i)}$ ;
                                    if  $f(\mathbf{p}_j^{(i)}) < f(\mathbf{g}^{(i)})$ 
                                        then {
                                             $\mathbf{g}^{(i)} \leftarrow \mathbf{p}_j^{(i)}$ ;
                                            if  $f(\mathbf{g}^{(i)}) < f(\mathbf{g})$ 
                                                then  $\mathbf{g} \leftarrow \mathbf{g}^{(i)}$ ;
                                        }
                                }
                        }
                    }
                }
            }
        }
    }
    Begin PSO: Level 2
    Initialize Swarm  $\mathbf{y}^{(i)} = \mathbf{g}^{(i)} : i = 1, 2, \dots, M$ ;
    for  $k \leftarrow 0$  to  $L_2$ 
        do {
            for  $i \leftarrow 1$  to  $M$ 
                do {
                     $\mathbf{v}^{(i)} \leftarrow \omega_2 \mathbf{v}^{(i)} + c_3 r_3 (\mathbf{p}^{(i)} - \mathbf{y}^{(i)}) + c_4 r_4 (\mathbf{g} - \mathbf{y}^{(i)})$ ;
                     $\mathbf{y}^{(i)} \leftarrow \mathbf{y}^{(i)} + \mathbf{v}^{(i)}$ ;
                    if  $f(\mathbf{y}^{(i)}) < f(\mathbf{p}^{(i)})$ 
                        do {
                             $\mathbf{p}^{(i)} \leftarrow \mathbf{y}^{(i)}$ ;
                            if  $f(\mathbf{p}^{(i)}) < f(\mathbf{g})$ 
                                then  $\mathbf{g} \leftarrow \mathbf{p}^{(i)}$ ;
                        }
                }
            }
        }
    }
    Return  $\mathbf{g}, f(\mathbf{g})$ 

```

**LEVEL 1**

# Algoritmo

Level 1 Initialization:  $\mathbf{x}_j^{(i)}, \mathbf{v}_j^{(i)}, \mathbf{p}_j^{(i)}, \mathbf{g}^{(i)}$  within  $[x_{min}, x_{max}]^D$  randomly.

Level 2 Initialization:  $\mathbf{v}^{(i)}, \mathbf{p}^{(i)}, \mathbf{g}$  within  $[x_{min}, x_{max}]^D$  randomly.

for  $EP \leftarrow 1$  to  $EP_{max}$

```

do {
    Begin PSO: Level 1
    for  $i \leftarrow 1$  to  $M$ 
        do {
            for  $k \leftarrow 0$  to  $L_1$ 
                do {
                    for  $j \leftarrow 1$  to  $N$ 
                        do {
 $\mathbf{v}_j^{(i)} \leftarrow \omega_1 \mathbf{v}_j^{(i)} + c_1 r_1 (\mathbf{p}_j^{(i)} - \mathbf{x}_j^{(i)}) + c_2 r_2 (\mathbf{g}^{(i)} - \mathbf{x}_j^{(i)});$ 
 $\mathbf{x}_j^{(i)} \leftarrow \mathbf{x}_j^{(i)} + \mathbf{v}_j^{(i)};$ 
                        if  $f(\mathbf{x}_j^{(i)}) < f(\mathbf{p}_j^{(i)})$ 
                            do {
                                 $\mathbf{p}_j^{(i)} \leftarrow \mathbf{x}_j^{(i)};$ 
                                if  $f(\mathbf{p}_j^{(i)}) < f(\mathbf{g}^{(i)})$ 
                                    then {
                                         $\mathbf{g}^{(i)} \leftarrow \mathbf{p}_j^{(i)};$ 
                                        if  $f(\mathbf{g}^{(i)}) < f(\mathbf{g})$ 
                                            then  $\mathbf{g} \leftarrow \mathbf{g}^{(i)};$ 
                                    }
                            }
                        }
                    }
                }
            }
        }
    }
    Begin PSO: Level 2
    Initialize Swarm  $\mathbf{y}^{(i)} = \mathbf{g}^{(i)} : i = 1, 2, \dots, M;$ 
    for  $k \leftarrow 0$  to  $L_2$ 
        do {
            for  $i \leftarrow 1$  to  $M$ 
                do {
 $\mathbf{v}^{(i)} \leftarrow \omega_2 \mathbf{v}^{(i)} + c_3 r_3 (\mathbf{p}^{(i)} - \mathbf{y}^{(i)}) + c_4 r_4 (\mathbf{g} - \mathbf{y}^{(i)});$ 
 $\mathbf{y}^{(i)} \leftarrow \mathbf{y}^{(i)} + \mathbf{v}^{(i)};$ 
                if  $f(\mathbf{y}^{(i)}) < f(\mathbf{p}^{(i)})$ 
                    do {
                         $\mathbf{p}^{(i)} \leftarrow \mathbf{y}^{(i)};$ 
                        if  $f(\mathbf{p}^{(i)}) < f(\mathbf{g})$ 
                            then  $\mathbf{g} \leftarrow \mathbf{p}^{(i)};$ 
                    }
                }
            }
        }
    }
    Return  $\mathbf{g}, f(\mathbf{g})$ 

```

**LEVEL 1**

$$\omega_1 = 1 - \frac{k}{L_1 + 1}$$

# Algoritmo

Level 1 Initialization:  $\mathbf{x}_j^{(i)}, \mathbf{v}_j^{(i)}, \mathbf{p}_j^{(i)}, \mathbf{g}^{(i)}$  within  $[x_{min}, x_{max}]^D$  randomly.

Level 2 Initialization:  $\mathbf{v}^{(i)}, \mathbf{p}^{(i)}, \mathbf{g}$  within  $[x_{min}, x_{max}]^D$  randomly.

for  $EP \leftarrow 1$  to  $EP_{max}$

**LEVEL 1**

Begin PSO: Level 1

for  $i \leftarrow 1$  to  $M$

do {

for  $k \leftarrow 0$  to  $L_1$

do {

for  $j \leftarrow 1$  to  $N$

do {

$\mathbf{v}_j^{(i)} \leftarrow \omega_1 \mathbf{v}_j^{(i)} + c_1 r_1 (\mathbf{p}_j^{(i)} - \mathbf{x}_j^{(i)}) + c_2 r_2 (\mathbf{g}^{(i)} - \mathbf{x}_j^{(i)});$

$\mathbf{x}_j^{(i)} \leftarrow \mathbf{x}_j^{(i)} + \mathbf{v}_j^{(i)};$

if  $f(\mathbf{x}_j^{(i)}) < f(\mathbf{p}_j^{(i)})$

then {

$\mathbf{p}_j^{(i)} \leftarrow \mathbf{x}_j^{(i)};$

if  $f(\mathbf{p}_j^{(i)}) < f(\mathbf{g}^{(i)})$

then {

$\mathbf{g}^{(i)} \leftarrow \mathbf{p}_j^{(i)};$

if  $f(\mathbf{g}^{(i)}) < f(\mathbf{g})$

then  $\mathbf{g} \leftarrow \mathbf{g}^{(i)};$

}

}

}

}

}

do {

Begin PSO: Level 2

Initialize Swarm  $\mathbf{y}^{(i)} = \mathbf{g}^{(i)} : i = 1, 2, \dots, M;$

for  $k \leftarrow 0$  to  $L_2$

do {

for  $i \leftarrow 1$  to  $M$

do {

$\mathbf{v}^{(i)} \leftarrow \omega_2 \mathbf{v}^{(i)} + c_3 r_3 (\mathbf{p}^{(i)} - \mathbf{y}^{(i)}) + c_4 r_4 (\mathbf{g} - \mathbf{y}^{(i)});$

$\mathbf{y}^{(i)} \leftarrow \mathbf{y}^{(i)} + \mathbf{v}^{(i)};$

if  $f(\mathbf{y}^{(i)}) < f(\mathbf{p}^{(i)})$

then {

$\mathbf{p}^{(i)} \leftarrow \mathbf{y}^{(i)};$

if  $f(\mathbf{p}^{(i)}) < f(\mathbf{g})$

then  $\mathbf{g} \leftarrow \mathbf{p}^{(i)};$

}

}

}

do {

Return  $\mathbf{g}, f(\mathbf{g})$

coeficientes de  
aceleración

**LEVEL 1**

$c_1 = c_2 = 2.05$

# Algoritmo

Level 1 Initialization:  $\mathbf{x}_j^{(i)}, \mathbf{v}_j^{(i)}, \mathbf{p}_j^{(i)}, \mathbf{g}^{(i)}$  within  $[x_{min}, x_{max}]^D$  randomly.

Level 2 Initialization:  $\mathbf{v}^{(i)}, \mathbf{p}^{(i)}, \mathbf{g}$  within  $[x_{min}, x_{max}]^D$  randomly.

for  $EP \leftarrow 1$  to  $EP_{max}$

```

do {
  Begin PSO: Level 1
  for  $i \leftarrow 1$  to  $M$ 
    do {
      for  $k \leftarrow 0$  to  $L_1$ 
        do {
          for  $j \leftarrow 1$  to  $N$ 
            do {
               $\mathbf{v}_j^{(i)} \leftarrow \omega_1 \mathbf{v}_j^{(i)} + c_1 r_1 (\mathbf{p}_j^{(i)} - \mathbf{x}_j^{(i)}) + c_2 r_2 (\mathbf{g}^{(i)} - \mathbf{x}_j^{(i)});$ 
               $\mathbf{x}_j^{(i)} \leftarrow \mathbf{x}_j^{(i)} + \mathbf{v}_j^{(i)};$ 
              if  $f(\mathbf{x}_j^{(i)}) < f(\mathbf{p}_j^{(i)})$ 
                do {
                   $\mathbf{p}_j^{(i)} \leftarrow \mathbf{x}_j^{(i)};$ 
                  if  $f(\mathbf{p}_j^{(i)}) < f(\mathbf{g}^{(i)})$ 
                    then {
                       $\mathbf{g}^{(i)} \leftarrow \mathbf{p}_j^{(i)};$ 
                      if  $f(\mathbf{g}^{(i)}) < f(\mathbf{g})$ 
                        then  $\mathbf{g} \leftarrow \mathbf{g}^{(i)}.$ 
                    }
                }
            }
          }
        }
      }
    }
  }

```

Begin PSO: Level 2

Initialize Swarm  $\mathbf{y}^{(i)} = \mathbf{g}^{(i)} : i = 1, 2, \dots, M;$

for  $k \leftarrow 0$  to  $L_2$

```

do {
  for  $i \leftarrow 1$  to  $M$ 
    do {
       $\mathbf{v}^{(i)} \leftarrow \omega_2 \mathbf{v}^{(i)} + c_3 r_3 (\mathbf{p}^{(i)} - \mathbf{y}^{(i)}) + c_4 r_4 (\mathbf{g} - \mathbf{y}^{(i)});$ 
       $\mathbf{y}^{(i)} \leftarrow \mathbf{y}^{(i)} + \mathbf{v}^{(i)};$ 
      if  $f(\mathbf{y}^{(i)}) < f(\mathbf{p}^{(i)})$ 
        do {
           $\mathbf{p}^{(i)} \leftarrow \mathbf{y}^{(i)};$ 
          then {
            if  $f(\mathbf{p}^{(i)}) < f(\mathbf{g})$ 
              then  $\mathbf{g} \leftarrow \mathbf{p}^{(i)};$ 
          }
        }
      }
    }
  }

```

**LEVEL 2**

Return  $\mathbf{g}, f(\mathbf{g})$



- 1 Introducción
- 2 Particle swarm optimization
- 3 Galaxy swarm optimization
- 4 Algoritmo
- 5 Código**

```
for(int ep = 0; ep < ep_max; ep++){
    // PSO level 1
    for(int i = 0; i < M; i++){
        for(int k = 0; k <= L1; k++){

            float w1 = 1.0-k*L1_div;

            for(int j = 0; j < N; j++){

                float r1 = Randfloat(0,1.0);
                float r2 = Randfloat(0,1.0);

                for(int d = 0; d < D; d++){
                    v1[i][j][d] = w1*v1[i][j][d] + c1*r1*(p1[i][j][d]-x[i][j][d])
                                     + c2*r2*(g[i][d]-x[i][j][d]);
                    x[i][j][d] = x[i][j][d] + v1[i][j][d];
                }

                if(f(x[i][j]) < f(p1[i][j])){
                    for(int d = 0; d < D; d++){
                        p1[i][j][d] = x[i][j][d];
                    }

                    if(f(p1[i][j]) < f(g[i])){
                        for(int d = 0; d < D; d++){
                            g[i][d] = p1[i][j][d];
                        }

                        if(f(g[i]) < f(gbest)){
                            for(int d = 0; d < D; d++){
                                gbest[d] = g[i][d];
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```
// PSO level 2
vector< vector<float> > y(g);

for(int k = 0; k <= L2; k++){
    float w2 = 1.0-k*1.0*L2_div;

    for(int i = 0; i < M; i++){
        float r3 = Randfloat(0,1.0);
        float r4 = Randfloat(0,1.0);

        for(int d = 0; d < D; d++){
            v2[i][d] = w2*v2[i][d] + c3*r3*(p2[i][d] - y[i][d])
                      + c4*r4*(gbest[d]-y[i][d]);
            y[i][d] = y[i][d] + v2[i][d];
        }

        if(f(y[i]) < f(p2[i])){
            for(int d = 0; d < D; d++)
                p2[i][d] = y[i][d];

            if(f(p2[i]) < f(gbest)){
                for(int d = 0; d < D; d++)
                    gbest[d] = p2[i][d];
            }
        }
    }
}

return gbest;
```

# Parámetros

Dimensión	M	N	$L_1$	$L_2$	$EP_{max}$	$c_1 = c_2 = c_3 = c_4$
10	10	5	198	1000	5	2.05
30	20	5	280	1500	5	2.05
50	20	5	250	1500	9	2.05

*FIN DE LA PRESENTACIÓN*