

## Algoritmo PSO [1]:

1. Iniciar  $t = 1$   
Inicializar cada partícula de la población en posiciones  $x_i$  y velocidades  $v_i$  aleatorias, para  $i = 1, 2, \dots, N_p$ .
2. Repetir hasta que se cumpla  $t = \text{MaxIt}$ :
  - a. Calcular el valor *fitness* para cada partícula  $i$   
Si el *fitness* de cada partícula  $i$  es mejor que su mejor solución, entonces actualizar  $x_i^g(t)$
  - b. Determinar la ubicación de la partícula con el mejor *fitness* y actualizar  $x^g(t)$  si es necesario
  - c. Para cada partícula  $i$ :  
Calcular su velocidad  $v_i(t+1)$   
Fin del ciclo
  - d. Actualizar la posición  $x_i$  de cada partícula  $i$
  - e. Aumentar  $t = t + 1$Fin del ciclo principal

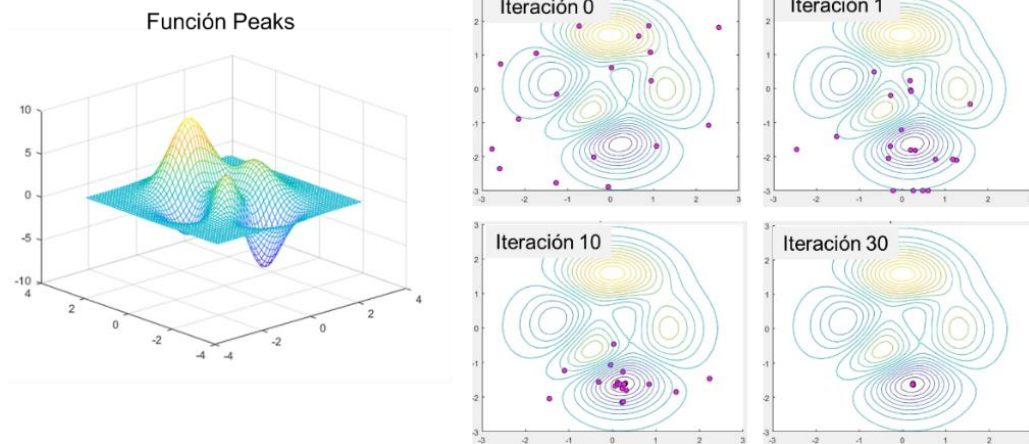
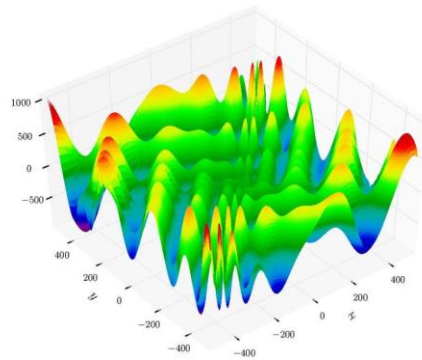


Figura 1. Se observa el comportamiento de las partículas en la función Peaks para diferentes iteraciones

## Ejemplo

Se optimiza la función Eggholder

$$f(x, y) = -(y + 47) \sin \sqrt{\left| \frac{x}{2} + (y + 47) \right|} - x \sin \sqrt{|x - (y + 47)|}$$



El mínimo se encuentra en

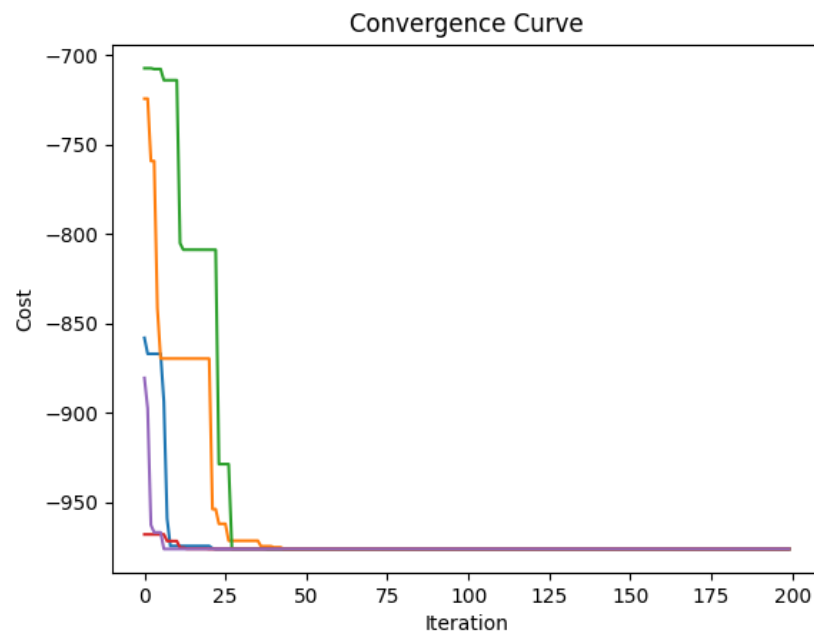
$$f(512, 404.2319) = -959.6407$$

$$-512 \leq x, y \leq 512$$

## Resultados

Encontró el mínimo [512, 404.236] en 5 de 5 corridas

Corrida	x1	x2	costo
1	520	411.385	-976.128
2	520	411.385	-976.128
3	520	411.385	-976.128
4	520	411.385	-976.128
5	520	411.385	-976.128



## Referencias

- [1] K.-L. Du and M. Swamy, Search and Optimization by Metaheuristics. Techniques and Algorithms Inspired by Nature, 1 ed., -: Birkhäuser Cham, DOI: <https://doi.org/10.1007/978-3-319-41192-7>, 2016.