

UNIVERSIDAD DEL VALLE DE GUATEMALA

Modelación y Simulación - CC2017

Sección 21

Ing. Luis Alberto Suriano



Excelencia que trasciende

DEL VALLE
GRUPO EDUCATIVO

Laboratorio No. 1

José Pablo Orellana 21970

Diego Alberto Leiva 21752

Gustavo Andrés González 21438

GUATEMALA, 16 de julio del 2024

Ejercicio 1

Responda las siguientes preguntas de forma clara

1. ¿En qué se diferencian los algoritmos de optimización genéticos y los de enjambre de partículas?

- Los algoritmos genéticos y los de enjambre, son técnicas de optimización basadas en la naturaleza pero cada uno tiene enfoques diferentes. Los algoritmos genéticos toman referencia en la evolución biológica, usando operadores lógicos como la selección, la cruce y la mutación para poder generar nuevas soluciones a partir de una población inicial. Mientras que los de enjambre se basan en el comportamiento social de los animales, donde las partículas, representando a las soluciones, se mueven en el espacio de búsqueda influenciadas por sus mejores posiciones personales y la mejor posición global del enjambre. (Mitchell, 1998) (Kennedy & Eberhart, 1995).

2. ¿En qué parte de los algoritmos de enjambre de partículas se considera la “exploración” y “explotación”?

- La exploración y explotación son aspectos de mucha importancia en los algoritmos de enjambre. La exploración se refiere a la capacidad del algoritmo para buscar en el espacio de soluciones globalmente y es promovida por componentes aleatorias que influyen en la velocidad de las partículas, permitiéndoles investigar nuevas áreas del espacio de búsqueda. Mientras que la explotación se enfoca en refinar las soluciones alrededor de las mejores posiciones encontradas hasta el momento. Esto se logra mediante el ajuste de las partículas hacia su mejor posición personal y la mejor posición global, lo que permite una búsqueda más detallada y precisa en esas áreas. (Shi & Eberhart, 1998)

3. ¿Se pueden paralelizar los algoritmos de enjambre de partículas?

- Los algoritmos de enjambre de partículas pueden ser paralelizados de manera efectiva debido a la independencia de las partículas durante cada iteración. Cada partícula puede actualizar su posición y velocidad en paralelo, lo que reduce significativamente el tiempo de cómputo y mejora la eficiencia del algoritmo. Además, la comunicación necesaria entre las partículas para compartir las mejores posiciones encontradas puede ser implementada de manera sincronizada, lo que facilita su implementación en sistemas paralelos y distribuidos. (Engelbrecht, 2005).

Ejercicio 2

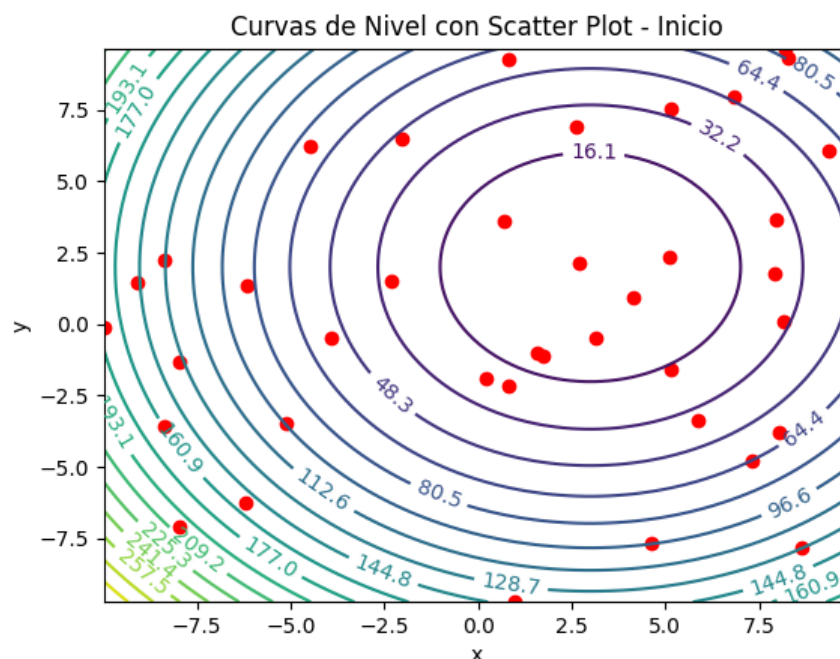
Mediante el uso de algoritmos de enjambre de partículas, resuelva el siguiente ejercicio.

Implemente un algoritmo de optimización de enjambre de partículas (PSO) para encontrar el mínimo de la función: $f(x, y) = (x - 3) ** 2 + (y - 2) ** 2$.

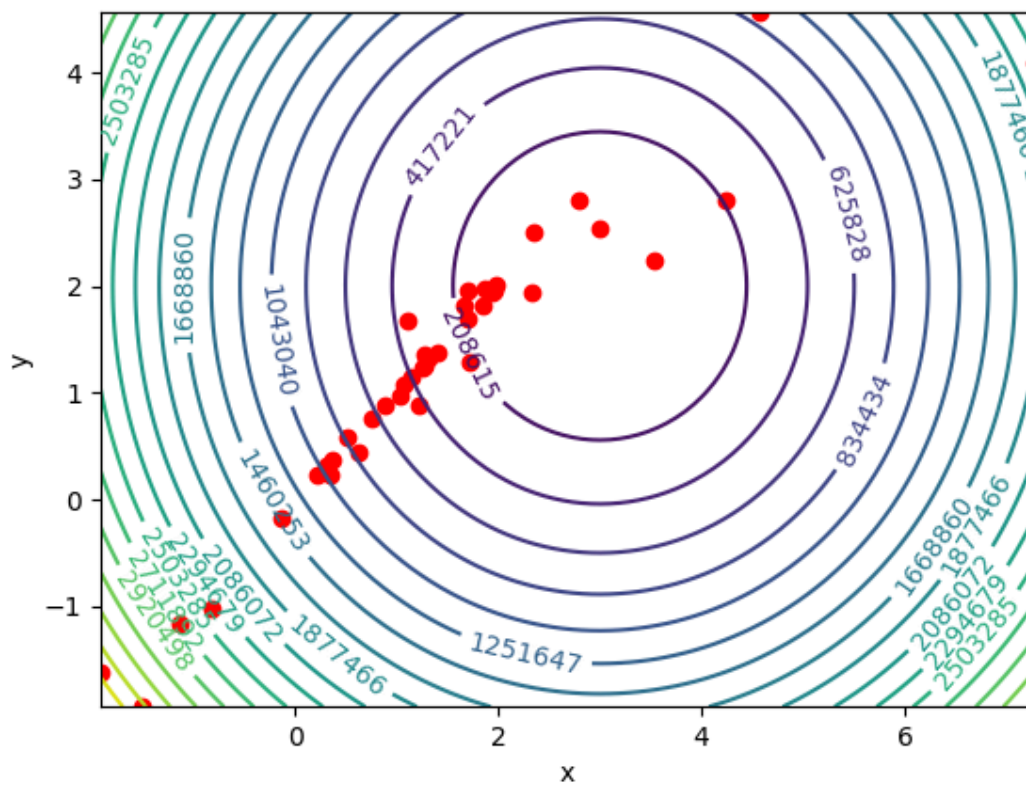
Para ello considere lo siguiente:

1. Inicialice el enjambre con $N = 40$ partículas
2. Cada posición de partículas estará dentro del rango $[-10, 10]$ tanto para x como y .
3. La velocidad deberá estar en el rango $[-1, 1]$
4. Establezca la mejor posición personal inicial de cada partícula en su posición inicial y la mejor posición global en la mejor posición inicial de partícula.
5. Para cada partícula
 - Actualice su velocidad según lo visto en clase
 - Actualice su posición según lo visto en clase
6. Use inicialmente $w=0.5$, $c1 = 1.5$, $c2 = 1.5$
7. $r1$ y $r2$ son valores aleatorios entre 0 y 1 (recuerde usar seed)
8. Actualice la mejor posición personal si la nueva posición tiene un valor de función más bajo que la mejor posición personal anterior.
9. Actualice la mejor posición global si la nueva posición tiene un valor de función más bajo que la mejor posición global anterior.
10. Repita este proceso a lo sumo 100 veces o hasta que el cambio en el global sea menor a un threshold definido por usted.

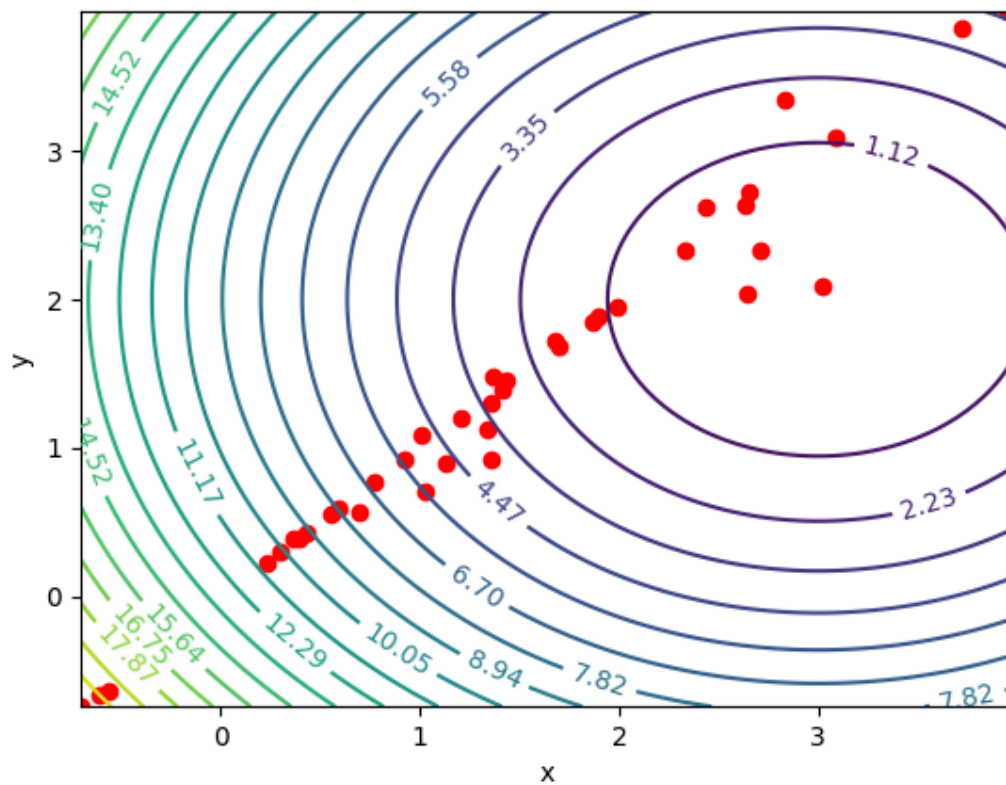
Recuerde graficar el contour plot de la función y mostrar la evolución de las partículas sobre este. Es decir, deberá mostrar al menos 3 gráficas del inicio, algún punto medio y el final de su iteración para encontrar la solución. Nota, la solución para esta función está en $(3,2)$.



Curvas de Nivel con Scatter Plot - Mitad

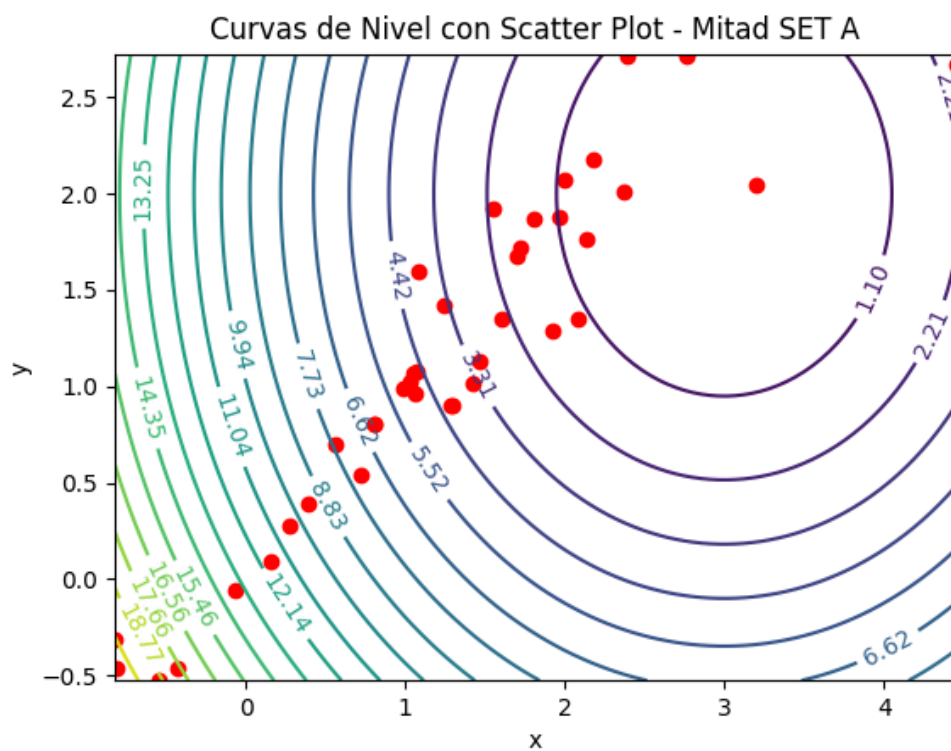
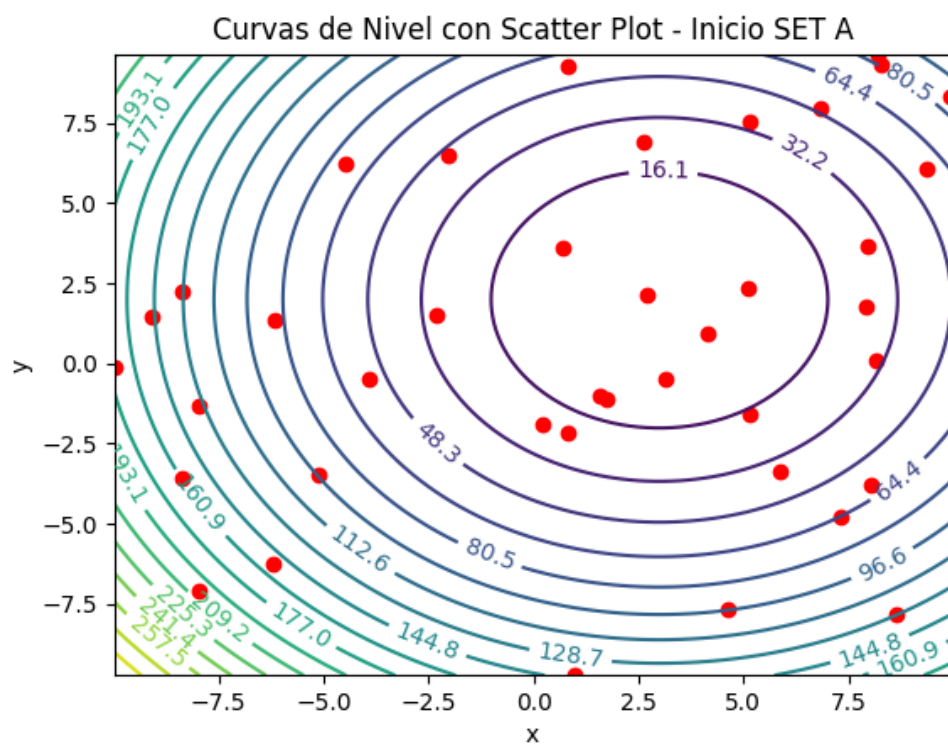


Curvas de Nivel con Scatter Plot - Final

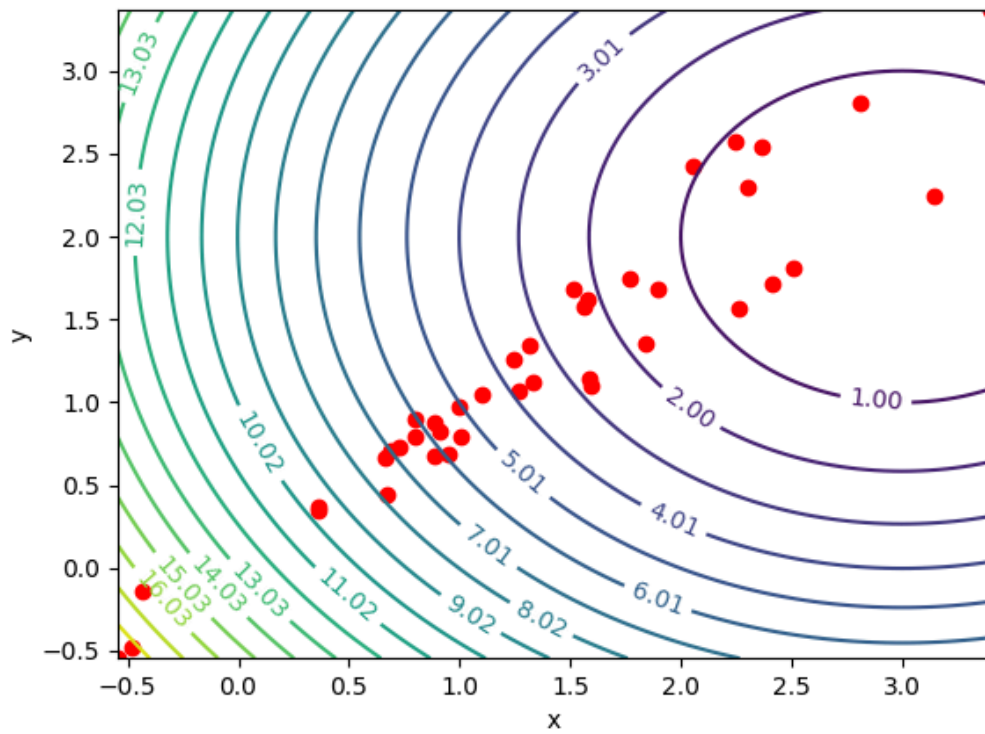


Ejercicio 3

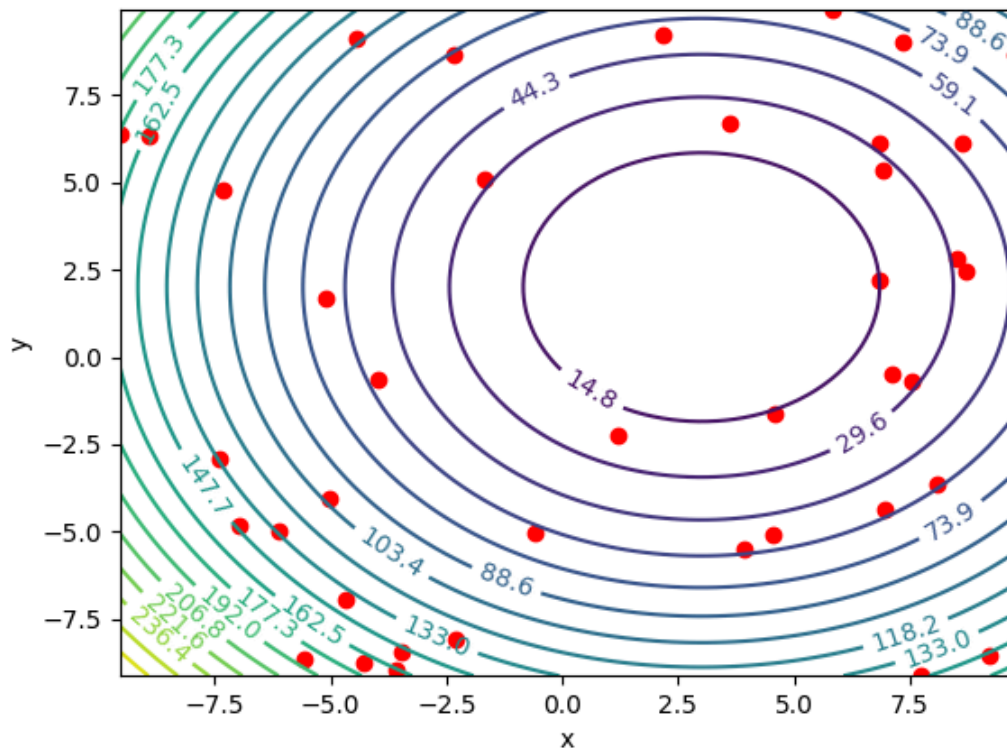
Usando el ejercicio anterior, pruebe cambiar los valores de w , $c1$, y $c2$ con nuevos valores que ustedes definan. Deben probar al menos 3 sets diferentes a los datos y sobre estos evaluar



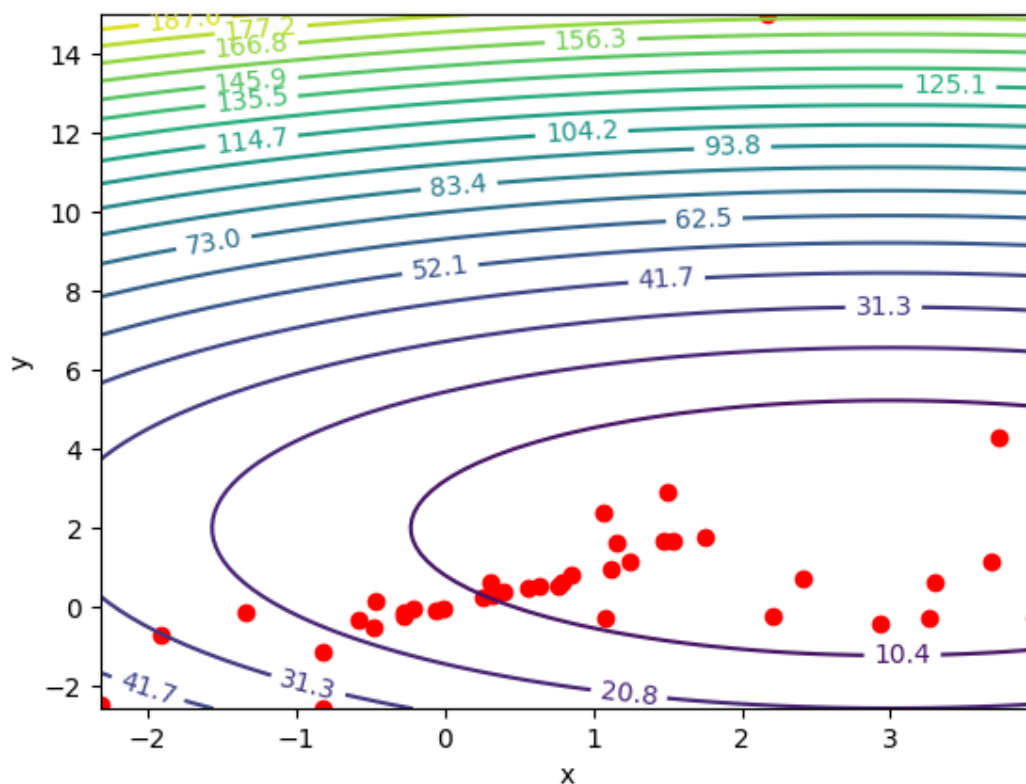
Curvas de Nivel con Scatter Plot - Final SET A



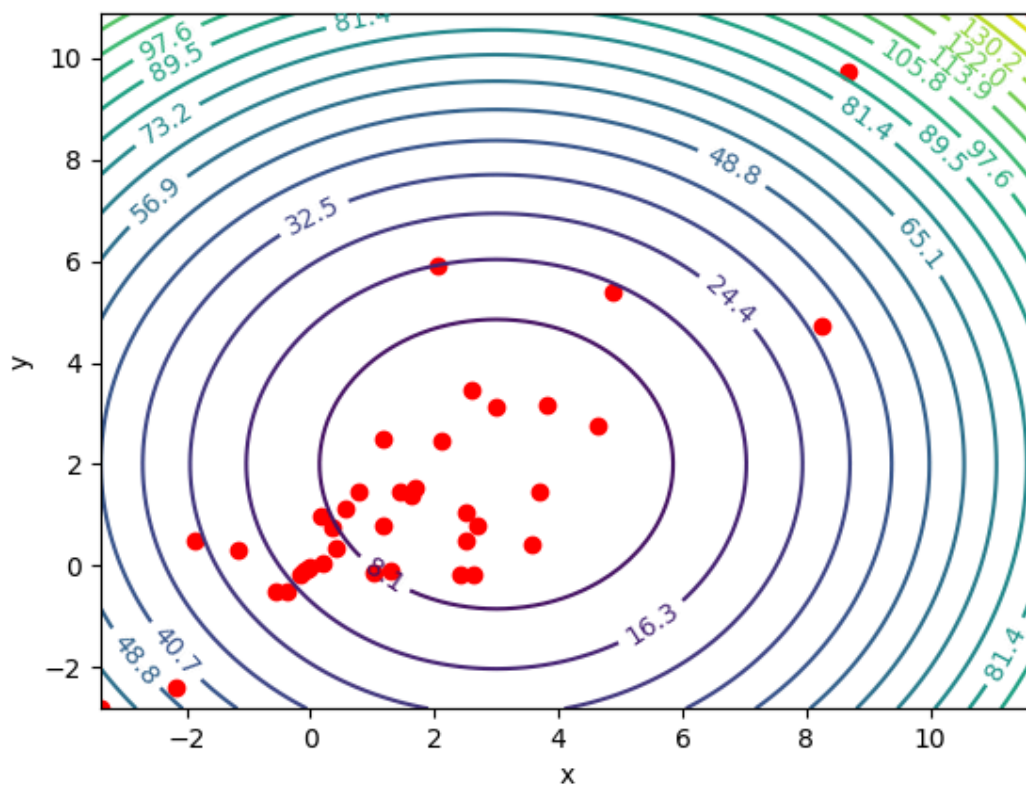
Curvas de Nivel con Scatter Plot - Inicio SET B



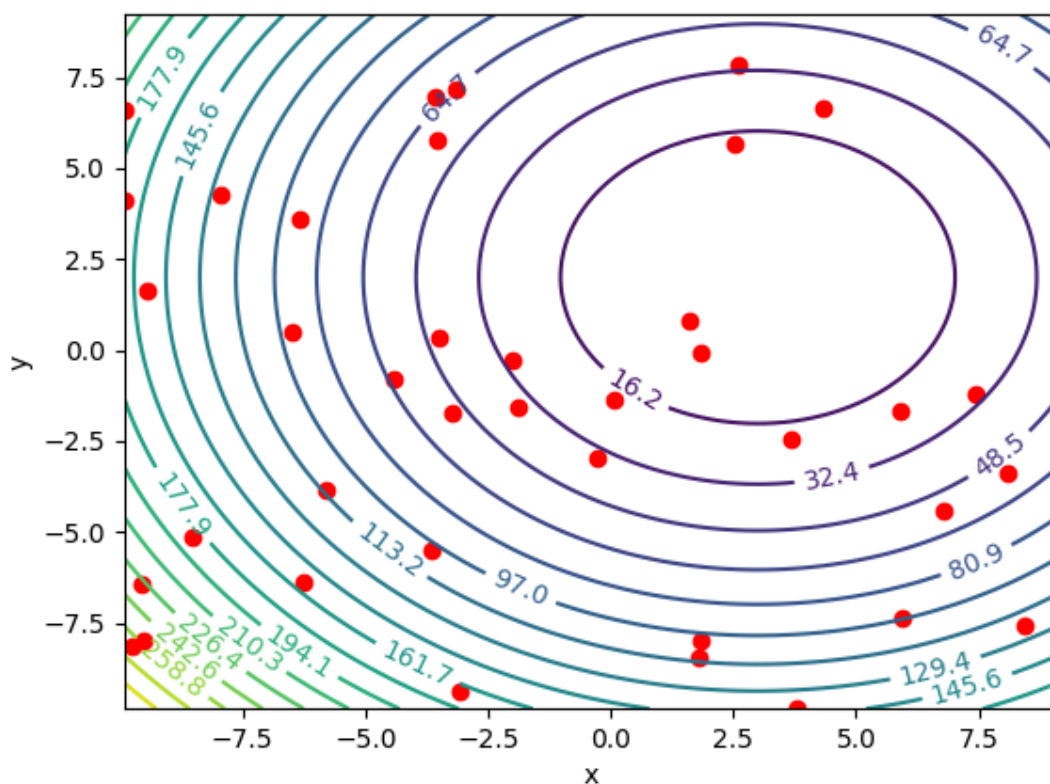
Curvas de Nivel con Scatter Plot - Mitad SET B



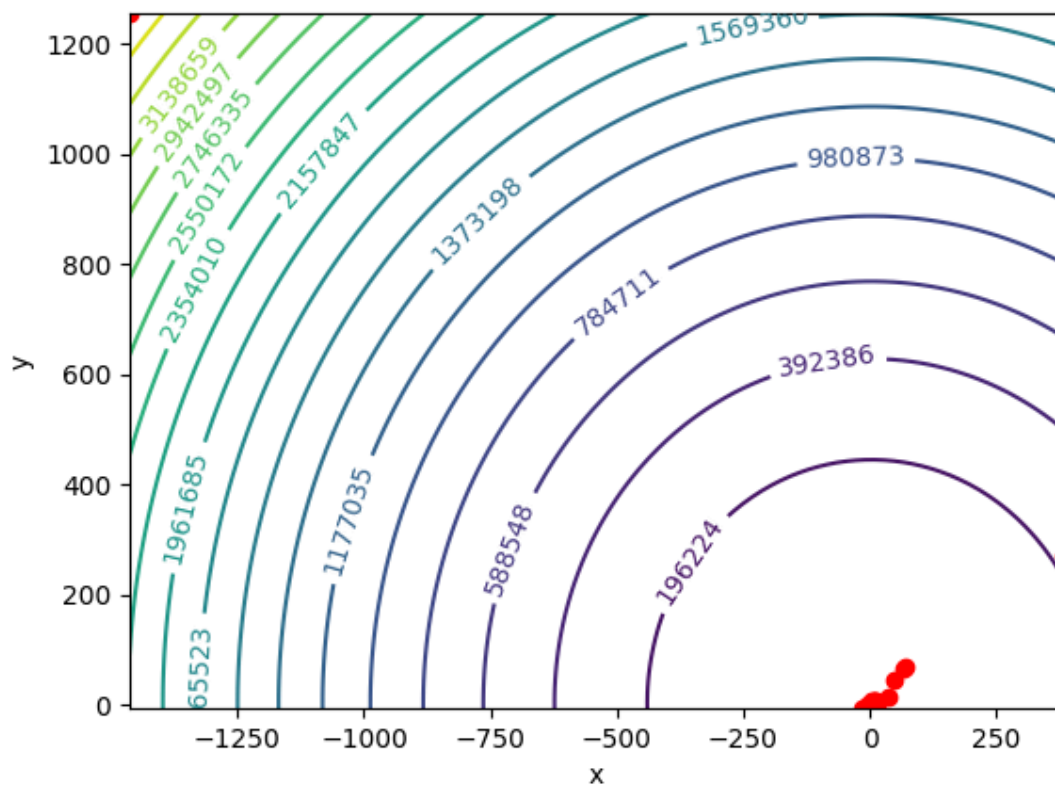
Curvas de Nivel con Scatter Plot - Final SET B

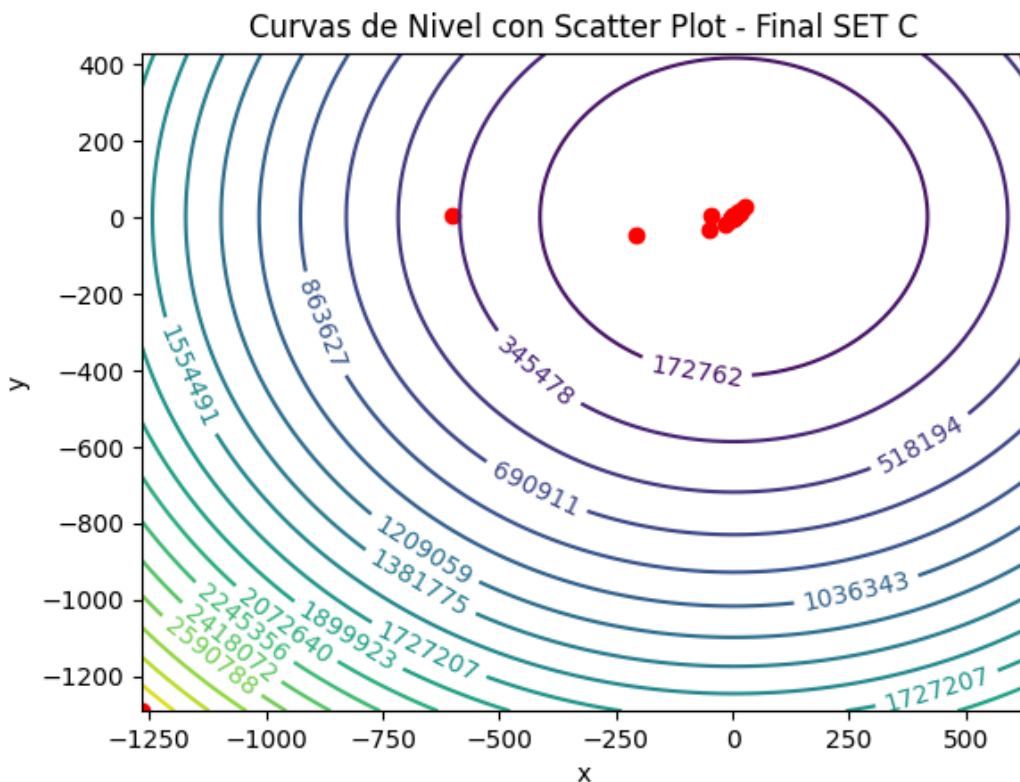


Curvas de Nivel con Scatter Plot - Inicio SET C



Curvas de Nivel con Scatter Plot - Mitad SET C





```
{'W': 0.4, 'C1': 1.3, 'C2': 1.4, 'label': 'SET A'},
{'W': 0.6, 'C1': 1.7, 'C2': 1.8, 'label': 'SET B'},
{'W': 0.8, 'C1': 2.0, 'C2': 2.1, 'label': 'SET C'}
```

```
Set: SET A
Best value: 0.0001800593202315259
Best position: [3.00439432 2.01267869]
Set: SET B
Best value: 0.0008491301009279328
Best position: [3.00249344 2.02903296]
Set: SET C
Best value: 0.0019159575594219036
Best position: [2.95653604 1.99481907]
```

1. ¿Cambió el resultado?

Si en cada set de parámetros hubo una ligera variación en comparación con el resultado del ejercicio anterior, pero las posiciones siguen siendo muy cercanas a (3,2), siendo el set C el que se desvía más del resultado esperado que los demás, pues sus valores de W, C1 Y C2 son los más grandes.

2. ¿Cuántas iteraciones tomó?

En los 3 sets diferentes para llegar al mejor resultado posible se tomaron 100 iteraciones.

Referencias

Engelbrecht, A. P. (2005). Fundamentals of Computational Swarm Intelligence. John Wiley & Sons.

Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks, 4, 1942-1948.

Mitchell, M. (1998). An Introduction to Genetic Algorithms. MIT Press.

Shi, Y., & Eberhart, R. (1998). A Modified Particle Swarm Optimizer. Proceedings of IEEE International Conference on Evolutionary Computation, 69-73.