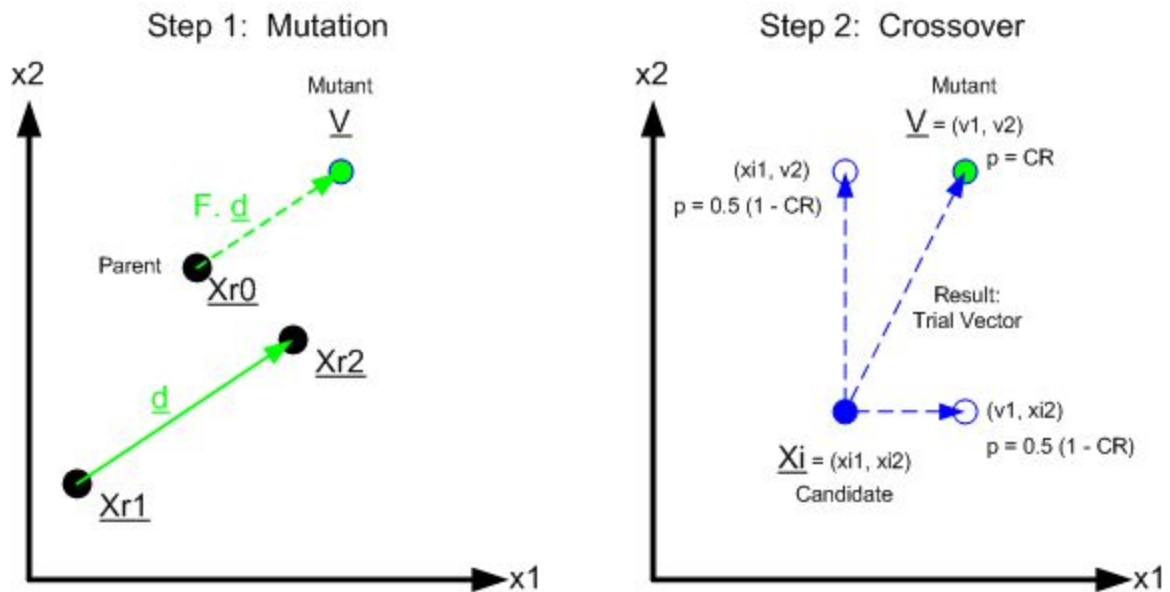


SSP. Inteligencia Artificial 1



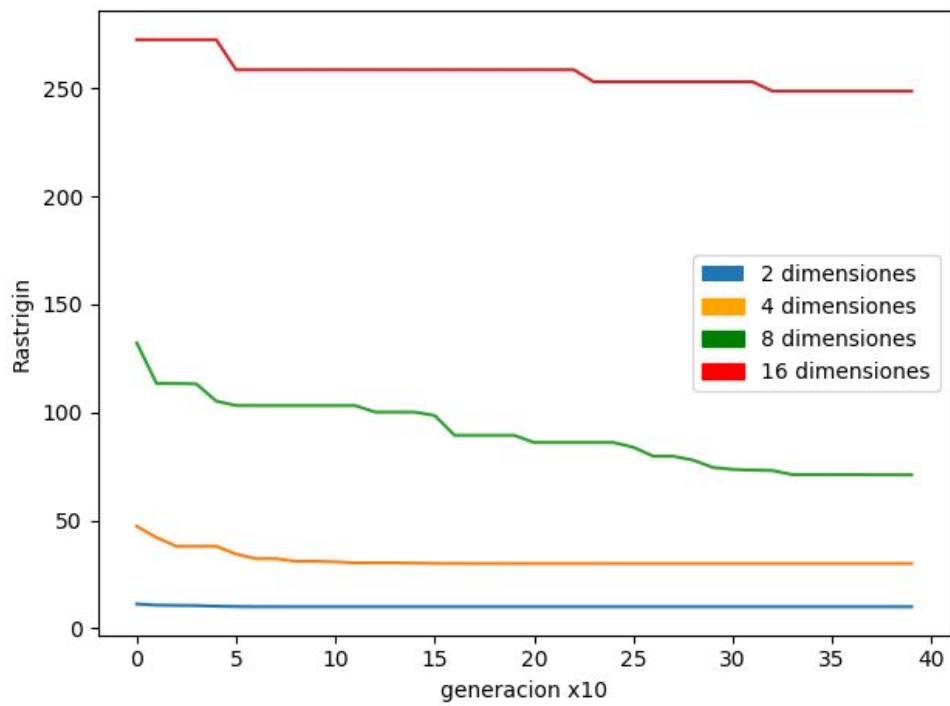
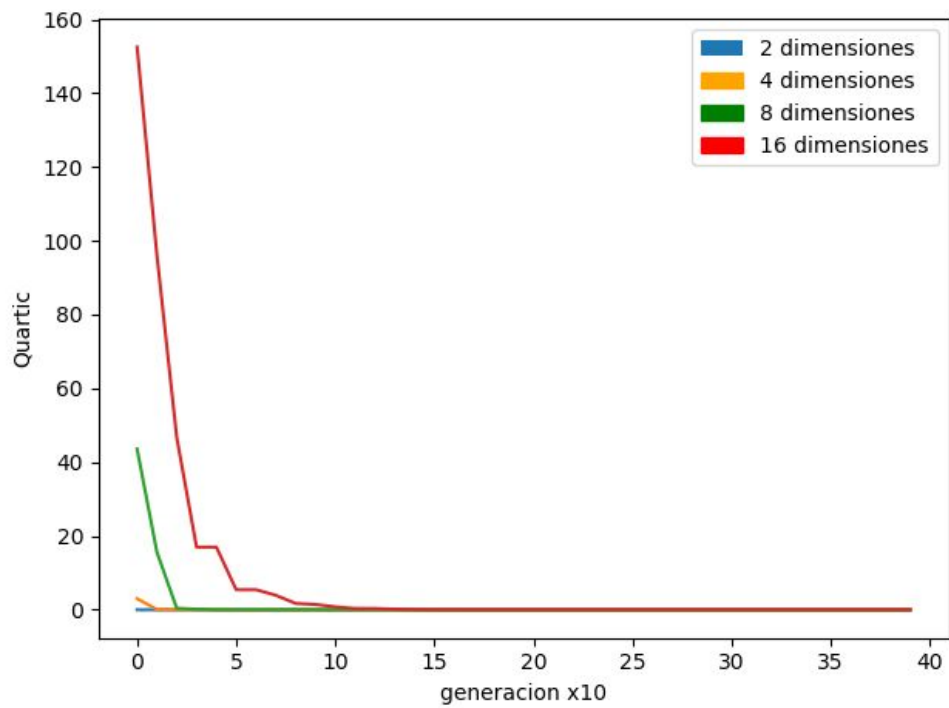
DE es un algoritmo basado en la población que está diseñado para optimizar funciones en un dominio continuo n -dimensional. Cada individuo en la población es un vector n dimensional que representa una solución candidata al problema.

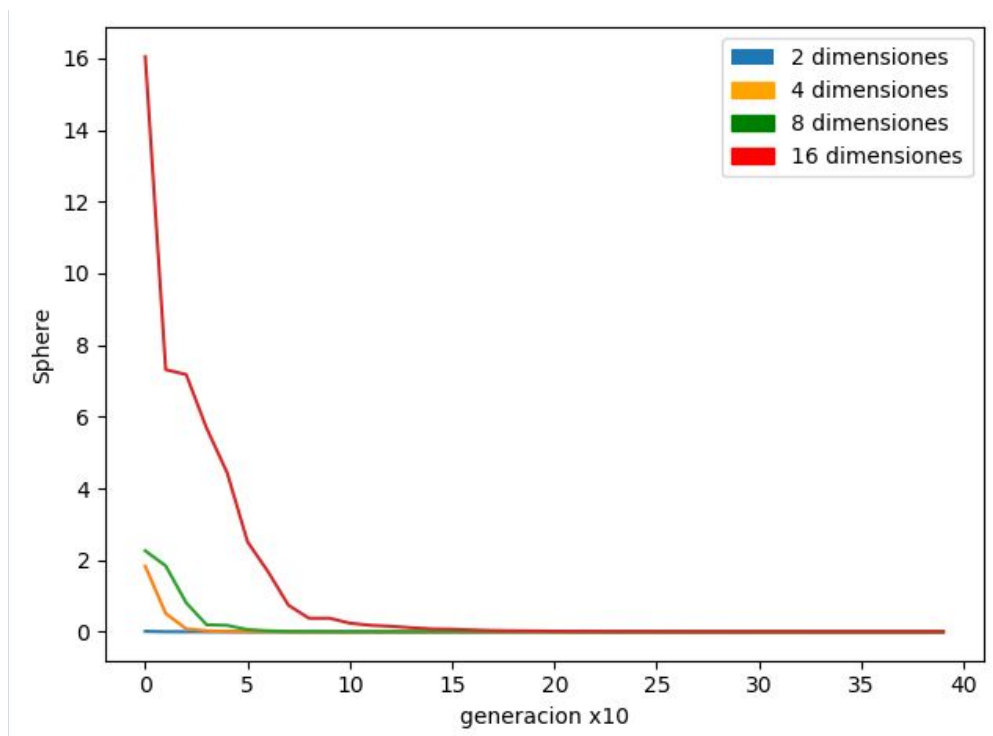
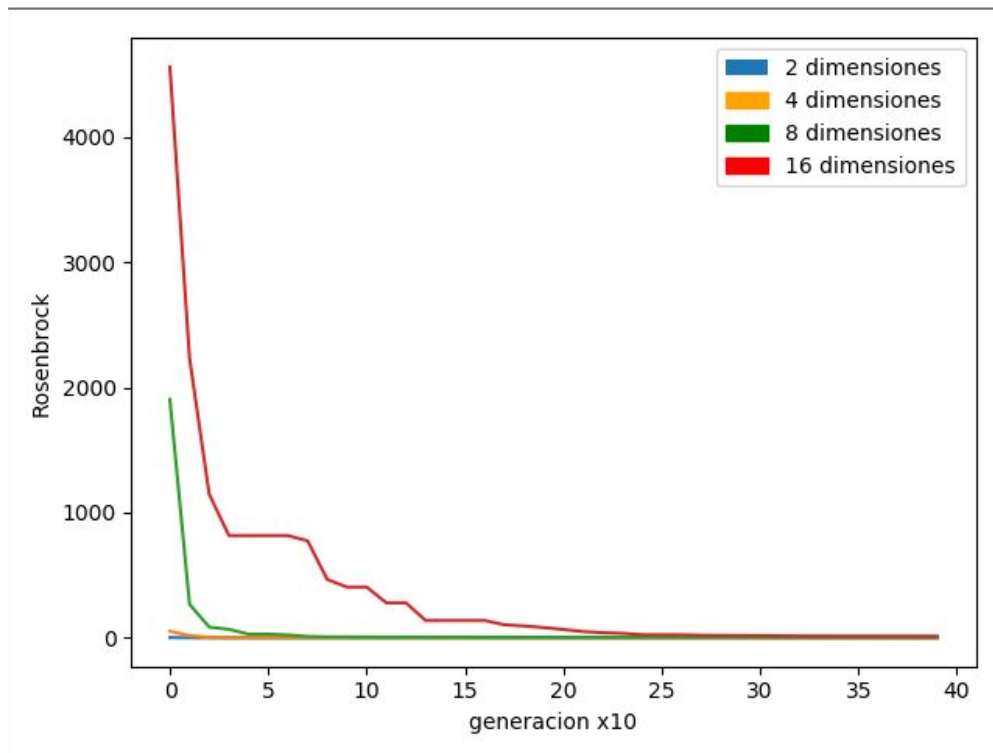
DE se basa en la idea de tomar el vector de diferencia entre dos individuos y agregar una versión escalada del vector de diferencia a un tercer individuo para crear una nueva solución candidata.

ED fue publicado por Kenneth Price y Rainer Storn en octubre de 1995 bajo el nombre de "Differential Evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces". Originalmente, el método estaba enfocado en la resolución del problema de ajuste de polinomio de Tchebychev utilizando una variante del método llamado Recocido Genético (Genetic Annealing), el cual había sido desarrollado por Price el año anterior. En 1996, ED fue presentado en el First International Contest on Evolutionary Optimization, que buscaba comparar el potencial de distintos métodos de optimización de computación evolutiva, finalizando ED en tercer lugar.

Hoy en día, ED representa una de las corrientes principales de la investigación en computación evolutiva.

Desarrollo de la actividad.





Conclusión:

El término Navaja de Occam es muy utilizado en software y dice que la solución más simple tiene más probabilidades de ser correcta(u óptima en este caso) que la compleja.

DE es exactamente eso, una abstracción del algoritmo AGC de una forma que ya no parece bioinspirado, visualiza los individuos como vectores n-dimensionales en un espacio de soluciones, y de acuerdo a su evaluación los corrige ligeramente y añade mutación

DE es muy eficiente en tiempo además y al igual que PSO es muy adecuado para trabajar con datos dinámicos y estocásticos. y esta dinámica verifica algunas propiedades de continuidad (el cambio no es muy brusco), entonces el mismo algoritmos hace que en cada iteración las partículas vayan evaluando el valor presente de la función a optimizar, por lo que pueden ir adaptando sus trayectorias a la situación real de la función.

Referencias:

K. Deb, Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, 2001.