***Trabajo Práctico Nº 1***

*Algoritmos Evolutivos (2024) – CEIA*

***Por:*** *Sevann Radhak Triztan*

*sevann.rahdak@gmail.com*

**Ejercicios.**

1. Mediante un algoritmo genético desarrollado en Python, encontrar el valor máximo de la función *y = x2*. Indicar en el informe el resultado de la solución encontrada (valor de “x”) si se ejecuta el algoritmo 10 lanzamientos. Los parámetros del algoritmo son:

* Selección por Ruleta
* Intervalo de la variable de decisión: [0, 31] ∈ Z
* Aplicar elitismo: Si
* Gen de cruza monopunto aleatorio
* Probabilidad de cruce 0.92
* Probabilidad de mutación 0.1
* Tamaño de la población: 4
* Generaciones: 10

**SOLUCION**

1. En este punto se describe la implementación y resultados obtenidos de un algoritmo genético diseñado para optimizar el valor de una función cuadrática *f(x)=x2*, donde *x* es un valor representado por un cromosoma binario:

**Implementación:**

El algoritmo utiliza selección por ruleta, cruce de un solo punto, y una tasa de mutación para evolucionar la población a lo largo de varias generaciones. Se aplicó un enfoque de elitismo para asegurar que los mejores individuos de una generación se mantuvieran en la población.

**Resultados:**

Los resultados de las 10 ejecuciones del algoritmo se resumen en la tabla a continuación, donde cada ejecución muestra el mejor individuo encontrado y su aptitud:

|  |  |  |
| --- | --- | --- |
| **Ejecución** | **Mejor solución *x*** | **Aptitud *y = x2*** |
| 1 | 31 | 961 |
| 2 | 30 | 900 |
| 3 | 30 | 900 |
| 4 | 30 | 900 |
| 5 | 31 | 961 |
| 6 | 31 | 961 |
| 7 | 30 | 900 |
| 8 | 31 | 961 |
| 9 | 31 | 961 |
| 10 | 30 | 900 |

**Conclusiones:**

El algoritmo genético logró encontrar soluciones óptimas consistentemente, con la mayoría de las ejecuciones convergiendo al valor máximo de *x=31*, que corresponde a la aptitud más alta de 961. Esto demuestra que la configuración utilizada fue efectiva para este problema en particular.

URL del repositorio donde se encuentra el algoritmo resuelto: