**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

**SOLUCIÓN:** 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.

*URL del repositorio donde se encuentra el algoritmo resuelto:* [*https://github.com/sevann-radhak/UBA-AE/blob/main/TP1/TP1-ej1.ipynb*](https://github.com/sevann-radhak/UBA-AE/blob/main/TP1/TP1-ej1.ipynb)

**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.

1. Minimizar mediante tres algoritmos genéticos desarrollados en Python la función y = x2.
   1. Indicar en el informe (en .pdf) el resultado de la solución encontrada (valor de “x”) si se ejecutan los 3 algoritmos un total de 30 lanzamientos cada uno. Los parámetros de los algoritmos son:

* Selección por Ranking, Ruleta y Torneo
* Intervalo de la variable de decisión: [-31, 31]∈R (con un dígito decimal)
* Aplicar elitismo: Si (solo en el método Ruleta y Ranking)
* Gen de cruza monopunto aleatorio
* Probabilidad de cruce 0.85
* Probabilidad de mutación 0.09
* Tamaño de la población: 4
* Generaciones: 10

**SOLUCIÓN:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lanzamiento** | **Solución Ranking** | **Solución Ruleta** | **Solución Torneo** |
| **1** | -0.3 | -10.5 | 4.1 |
| **2** | 0.3875 | 16.3 | -8.65625 |
| **3** | -1.3875 | 1.25 | 7.55 |
| **4** | 9.9 | 0.10000000000000053 | -5.1 |
| **5** | -0.2 | 0.20000000000000018 | -0.3500000000000001 |
| **6** | 4.7 | -2.3 | 9.7 |
| **7** | 0.5 | 0.05000000000000071 | 0.23749999999999982 |
| **8** | -0.05 | -10.3 | -3.2 |
| **9** | -2.2 | 5.9 | 0.3 |
| **10** | 0.1 | 10.5 | -0.5 |
| **11** | 0.2 | 6.9 | 0.8000000000000007 |
| **12** | -0.04375 | 4.6 | -0.0 |
| **13** | 0.004882812500000132 | 5.6 | 0.0652343749999998 |
| **14** | 0.275 | -2.4 | 6.125 |
| **15** | 4.4 | -6.4375 | 0.01562499999999999 |
| **16** | -0.275 | 2.0 | -0.7625 |
| **17** | -8.1 | -1.3 | 1.7 |
| **18** | -10.0 | 1.3499999999999996 | 8.0 |
| **19** | 0.1 | -6.8999999999999995 | -0.0 |
| **20** | -0.825 | 4.6 | 1.4 |
| **21** | -19.6 | -11.5 | 0.04999999999999999 |
| **22** | -3.5 | 1.2 | -7.35 |
| **23** | -0.0 | -2.7 | -12.2 |
| **24** | -0.11875 | 1.4 | 16.1 |
| **25** | 0.2 | -3.4 | 2.6 |
| **26** | -3.7 | -5.1 | 1.4500000000000002 |
| **27** | -6.5 | -0.09999999999999964 | -2.3499999999999996 |
| **28** | -3.608224830031759e-16 | -7.3 | 3.5 |
| **29** | 11.9 | 8.1 | 0.1 |
| **30** | 8.3 | -2.4 | 0.08749999999999994 |

* 1. Completar la siguiente tabla en base a las 30 ejecuciones con los parámetros señalados.

**SOLUCIÓN:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algoritmo** | **Mínimo** | **Promedio** | **Máximo** | **Desv. Est.** |
| **Ranking** | -19.2 | -1.9410481770833 | 7.3 | 5.352813230631482 |
| **Ruleta** | -12.2 | -0.8447916666667 | 6.83125 | 4.257650614914351 |
| **Torneo** | -13.85 | -0.6021875 | 14.9 | 6.233364303647925 |

* 1. Explicar una interpretación de los resultados obtenidos en el ítem anterior.

**SOLUCIÓN:** los resultados obtenidos a partir de los tres algoritmos genéticos (Ranking, Ruleta y Torneo) muestran diferencias en su capacidad para minimizar la función (*y = x2*).

* **Ranking:** obtuvo el mejor valor mínimo de -19.2, con un promedio de -1.94 y una desviación estándar de 5.35, lo que indica una mayor estabilidad y precisión en las soluciones.
* **Ruleta:** presentó un valor mínimo de -12.2, un promedio de -0.84 y una desviación estándar de 4.26, mostrando también una buena consistencia, aunque menos precisa que Ranking. Su promedio sugiere que el algoritmo tiende a encontrar soluciones más cercanas a cero en comparación con el Ranking.
* **Torneo:** sin elitismo, tuvo un valor mínimo de -13.85, un promedio de -0.60 y una desviación estándar de 6.23, indicando mayor variabilidad en los resultados y menor capacidad de conservar buenos individuos.
  1. Modificar los parámetros Pm, Tamaño de la población y Generaciones de modo tal que se consiga encontrar una combinación que permita obtener el mejor valor óptimo y su correspondiente solución (para cada algoritmo habrá una combinación diferente). Transcribir las combinaciones encontradas en el .pdf.

**SOLUCIÓN:** a continuación, se presentan los mejores parámetros encontrados para cada algoritmo:

* **Algoritmo Ranking:**
  + Mejor Solución: 0.0
  + Parámetros: (P\_m = 0.05), Tamaño de Población: 12, Generaciones: 10
* **Algoritmo Ruleta:**
  + Mejor Solución: 30.1516
  + Parámetros: (P\_m = 0.09), Tamaño de Población: 4, Generaciones: 20
* **Algoritmo Torneo:**
  + Mejor Solución: 0.0
  + Parámetros: (P\_m = 0.05), Tamaño de Población: 8, Generaciones: 10