

# Trabajo Práctico N° 1

## Algoritmos Evolutivos (2024) – CEIA

*Por: Sevann Radhak Triztan  
sevann.rahdak@gmail.com*

### EJERCICIOS.

- Mediante un algoritmo genético desarrollado en Python, encontrar el valor máximo de la función  $y = x^2$ . 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:  $[0, 31] \in \mathbb{Z}$
  - 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  $f(x)=x^2$ , 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 $y = x^2$ |
|-----------|--------------------|-------------------|
| 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: <https://github.com/sevann-radhak/UBA-AE/blob/main/TP1/TP1-ej1.ipynb>

2. Minimizar mediante tres algoritmos genéticos desarrollados en Python la función  $y = x^2$ .
- a. 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:  $[-31, 31] \in \mathbb{R}$  (con un dígito decimal)
  - Aplicar elitismo: Si (solo en el método Ruleta y Ranking)
  - Gen de cruce 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                  | 0.06523437499999998 |
| 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 |

- b. 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 |

- c. 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 ( $y = x^2$ ).

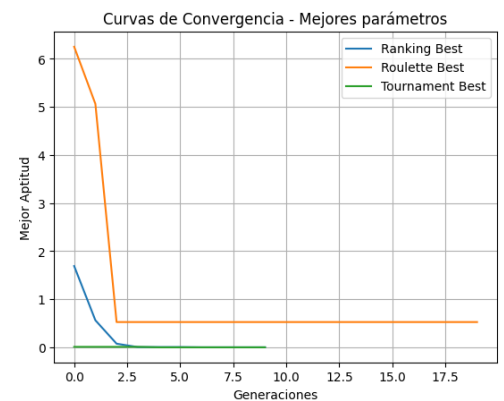
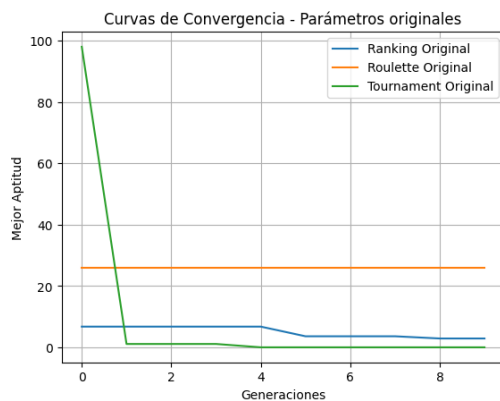
- **Ranking:** obtuvo el mejor valor mínimo de -19.2, con un promedio de -1.94 y una desviación estándar de 5.35, lo que indica una mayor estabilidad y precisión en las soluciones.

- **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.
- d. Modificar los parámetros  $P_m$ , 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
- e. Realizar 6 curvas de convergencia; 3 correspondientes a los algoritmos con los parámetros originales y 3 correspondientes a los algoritmos con los mejores parámetros encontrados en el ítem d. Mostrar las 6 curvas en el .pdf. Las curvas deben contener título, leyenda y etiquetas en los ejes.

**SOLUCIÓN:**



- f. Interpretar tanto las combinaciones de parámetros encontradas en cada uno de los 3 algoritmos como sus graficas de convergencia en el ítem anterior y explicarlas en el .pdf.

**SOLUCIÓN:**

- **Algoritmo Ranking:**
  - Generación 1: 2.25
  - Generación 10: 0.04
  - Interpretación: mejora significativa, alcanzando un valor bajo en la última generación.
- **Algoritmo Ruleta:**
  - Generación 1: 25.0
  - Generación 10: 1.69
  - Interpretación: estabilización en torno a 1.69, con dificultades para encontrar soluciones óptimas.
- **Algoritmo Torneo:**
  - Generación 1: 158.76

- Generación 10: 106.09
- Interpretación: alta inicial, pero sin convergencia efectiva a un valor bajo.

#### Mejores Parámetros Encontrados

- **Algoritmo Ranking:**
  - Generación 1: 19.36
  - Generación 10: 1.53e-07
  - Interpretación: convergencia hacia cero.
- **Algoritmo Ruleta:**
  - Generación 1: 441.0
  - Generación 20: 0.0225
  - Interpretación: gran mejora, estabilizándose en un valor muy bajo.
- **Algoritmo Torneo:**
  - Generación 1: 0.81
  - Generación 10: 0.0
  - Interpretación: convergencia efectiva a cero.

#### Conclusiones:

- Los ajustes de parámetros mejoraron notablemente el rendimiento.
- Ranking fue el más efectivo, seguido por Torneo y Ruleta.
- Las gráficas reflejan la capacidad de los algoritmos para converger hacia soluciones óptimas.

URL del repositorio donde se encuentra el algoritmo resuelto: <https://github.com/sevann-radhak/UBA-AE/blob/main/TP1/TP1-ej2.ipynb>

3. La distribución de la concentración de cierto contaminante en un canal está descrita por la ecuación

$$c(x,y) = 7.7 + 0.15x + 0.22y - 0.05x^2 - 0.016y^2 - 0.007xy$$

En donde, las variables independientes se encuentran entre los límites de  $-10 \leq x \leq 10$ ,  $0 \leq y \leq 20$ .

Para la función de adaptación anterior, escribir y ejecutar dos algoritmos genéticos que utilicen el operador de selección por ruleta y torneo respectivamente con probabilidades de cruce y mutación a elección. Luego realizar las siguientes consignas para ambos algoritmos:

- a. Determinar en forma aproximada la concentración máxima dada la función  $c(x, y)$ . Utilizar una precisión de 3 decimales. Transcribir en el .pdf el resultado obtenido en ambos algoritmos.

#### SOLUCIÓN:

- Algoritmo de Selección por Ruleta:
  - Mejor Solución: 8.361
- Algoritmo de Selección por Torneo:
  - Mejor Solución: 8.506

#### Conclusiones

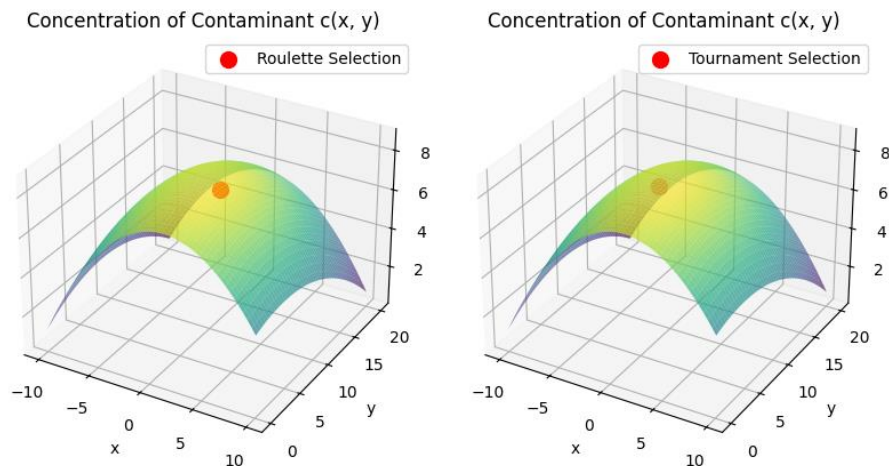
Ambos algoritmos lograron encontrar soluciones cercanas, con el algoritmo de selección por torneo alcanzando una mejor aptitud que el de selección por ruleta.

La diferencia en los resultados sugiere que la selección por torneo puede ser más efectiva en este contexto específico para maximizar la concentración del contaminante.

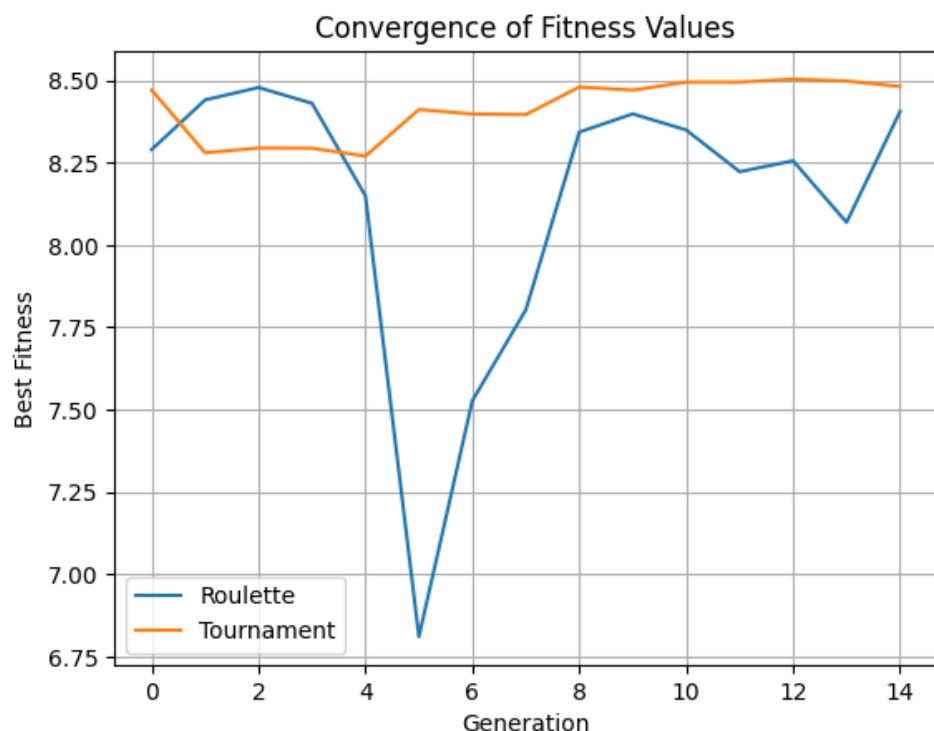
- b. Indicar la URL del repositorio (o URL Colab) donde se encuentra el algoritmo resuelto.

URL del repositorio donde se encuentra el algoritmo resuelto: <https://github.com/sevann-radhak/UBA-AE/blob/main/TP1/TP1-ej3.ipynb>

- c. Graficar  $c(x, y)$  en 3D para los intervalos de las variables independientes ya mencionados y agregar un punto rojo (ruleta) y un punto azul (torneo) en la gráfica en donde el algoritmo haya encontrado el valor máximo. Cada gráfico debe contener título, leyenda y etiquetas en los ejes.



- d. Graficar las mejores aptitudes encontradas en función de cada generación (Curva de convergencia de ambos algoritmos). Cada gráfico debe contener título, leyenda y etiquetas en los ejes.



- e. Realizar conclusiones/comentarios/observaciones respecto a los resultados obtenidos en ambos algoritmos.

#### SOLUCIÓN:

##### Evolución de la Aptitud:

- **Selección por Ruleta:** la mejor aptitud comenzó en 8.471 y finalizó en 8.361. Aunque hubo fluctuaciones, la tendencia general muestra que el algoritmo pudo encontrar soluciones competitivas, pero no logró mejorar significativamente en las últimas generaciones.
- **Selección por Torneo:** la mejor aptitud comenzó en 8.128 y finalizó en 8.506. Este algoritmo mostró una mejora constante a lo largo de las generaciones, alcanzando su mejor aptitud en la última generación.

##### Rendimiento Comparativo:

El algoritmo de selección por torneo superó al de selección por ruleta, alcanzando una mejor solución final. Esto sugiere que la selección por torneo es más eficaz en este contexto para maximizar la concentración del contaminante.

##### Consistencia:

La selección por torneo mostró una mayor consistencia en las aptitudes a lo largo de las generaciones, con menos variabilidad en los valores. Esto puede indicar que este método de selección favorece la convergencia hacia soluciones óptimas más rápidamente.