



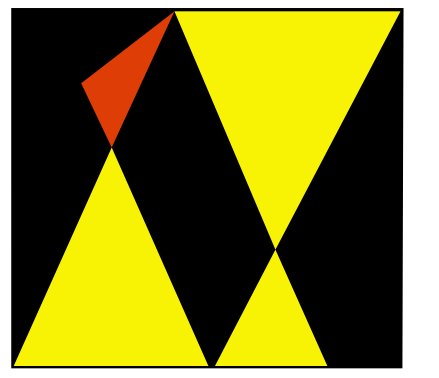
Universidad Veracruzana

Algoritmos Genéticos y el Problema de las 8 Reinas

Bryan Axel Pérez González

Facultad de Matemáticas, Universidad Veracruzana

zs19014432@estudiantes.uv.mx



1. Introducción

Este póster aborda el problema de las 8 reinas utilizando algoritmos genéticos, una técnica útil en la optimización combinatoria. El desafío consiste en posicionar ocho reinas en un tablero de ajedrez sin amenazarse mutuamente. Este trabajo busca no solo encontrar soluciones eficientes sino también resaltar la aplicabilidad de los algoritmos genéticos en problemas clásicos de combinatoria. El código del proyecto puede ser consultado en https://github.com/axel-perezg/geneticos_y_reinas

2. Problema

El problema consiste en colocar 8 reinas en un tablero de ajedrez regular de 8×8 de manera en que ningún par de reinas este bajo amenaza de ataque. La resolución de este problema mediante algoritmos genéticos implica el uso de técnicas basadas en la evolución biológica para encontrar una disposición óptima que cumpla con las restricciones del problema.

Para ello, hacemos uso de las siguientes herramientas.

Representación	Permutación
Inicialización	Aleatoria
Selección de padres	Universal Estocástica
Cruza	A los extremos
Mutación	Intercambio
Remplazo	Brecha generacional

Representación

Para aplicar el algoritmo genético, necesitamos expresar un tablero de forma adecuada. Podemos expresar por medio de permutaciones, es decir, un *array* de enteros de modo que los elementos representen las filas en las que esta cada reina. Así, el *array* $[0, 2, 4, 6, 1, 3, 5, 7]$ representa al tablero

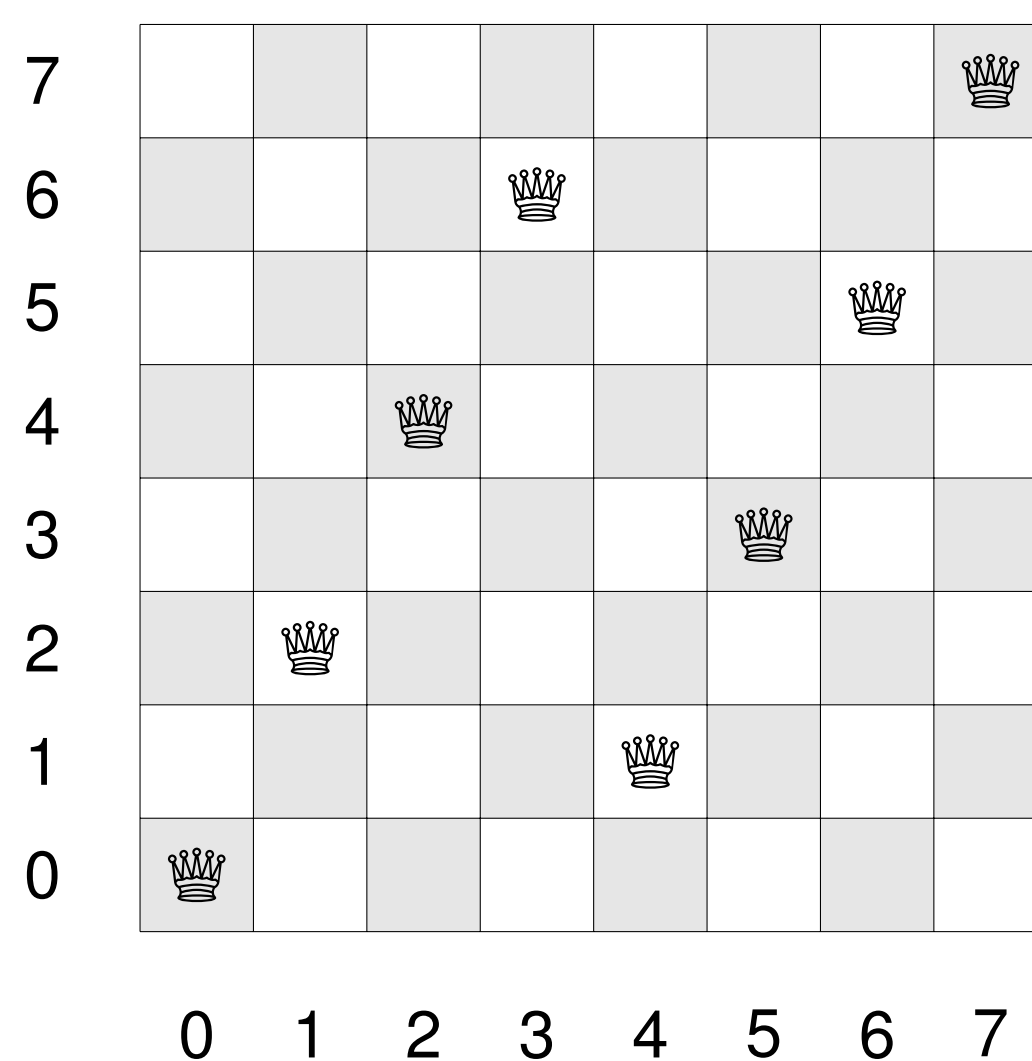


Figura 1: Tablero de ajedrez con reinas.

Inicialización

El algoritmo comienza con un conjunto k de individuos, en este caso *arrays* que representan tableros, generados aleatoriamente. Esta sería nuestra primer población.

Para poder determinar que tan idóneo es un tablero como solución del problema se hace uso de la función de aptitud (fitness).

En este caso, la función devuelve el número de pares de reinas que **no** se atacan entre sí. Dado que el número máximo de pares posibles es 28, mientras mayor sea el valor de la función, mas idónea será la solución al problema.

```
def fitness(tablero):
    ataques = sum(1 for i in range(8)
                  for j in range(i + 1, 8) if
                  abs(tablero[i] - tablero[j])
                  == abs(i - j))
    return 28 - ataques
```

Selección de padres

Luego para seleccionar los padres, hacemos uso del selección universal estocástica. A diferencia de la selección por ruleta, en vez de seleccionar un individuo y girar la rueda varias veces hasta seleccionar todos los individuos, en este caso la rueda se gira una vez y se seleccionan todos los individuos igualmente separados alrededor de la rueda. Se puede observar la idea en la siguiente figura.

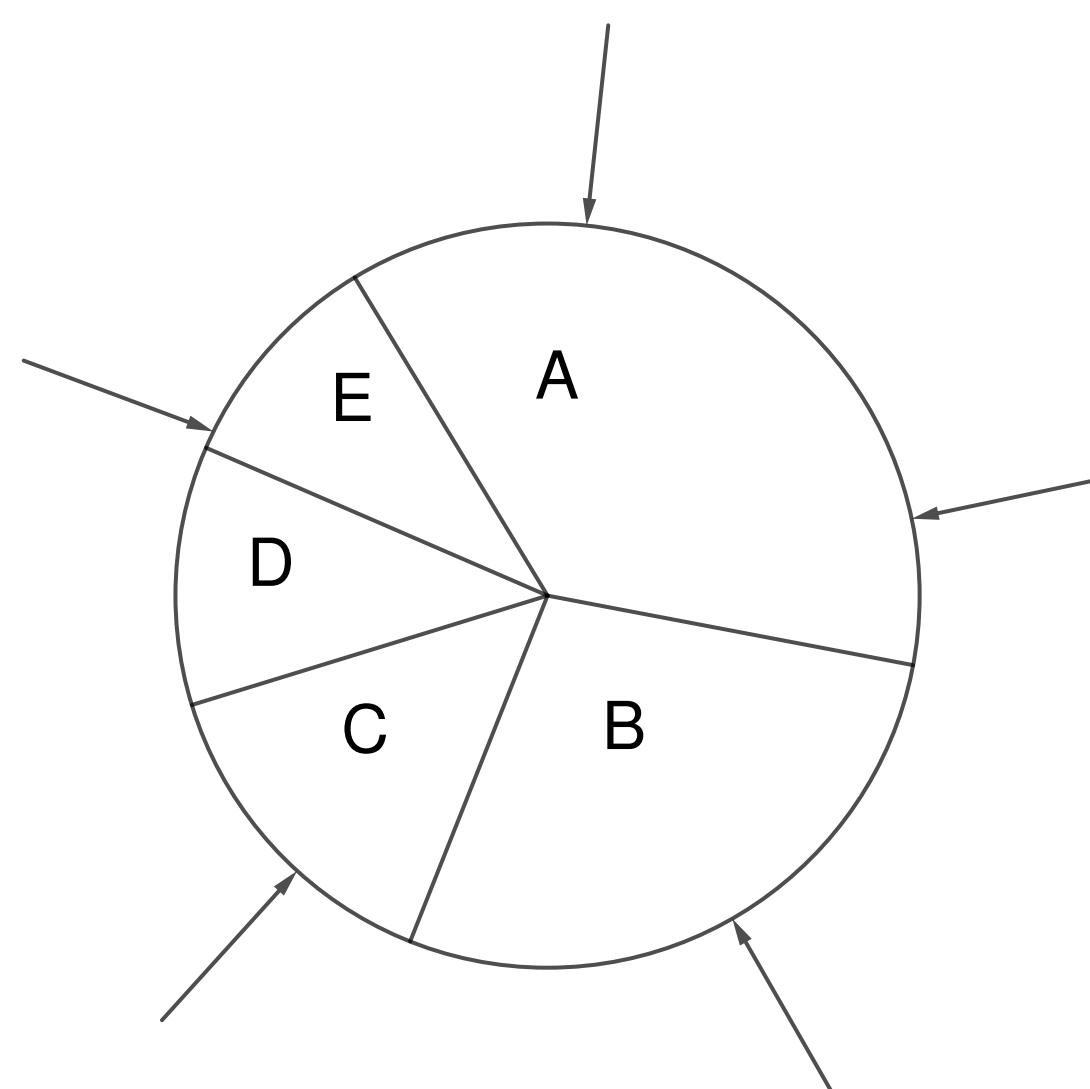


Figura 2: Selección Universal Estocástica

En el caso de la figura, se seleccionan los padres: A, A, B, C, E

Cruza

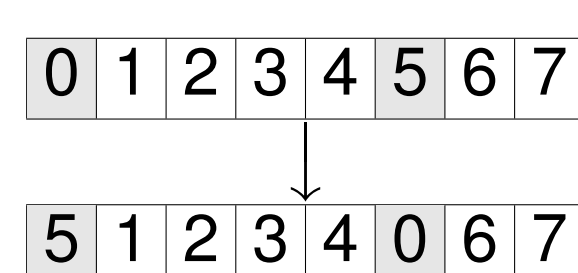
La cruce de extremos esta basada en la idea de que los descendientes deben ser creados, en medida de lo posible, utilizando solo los extremos que están presentes en los padres. Para esto, se construye una tabla de extremos, que para cada elemento enumera los otros elementos que están vinculados a él en los dos padres.

Si consideramos a los individuos $[0, 1, 2, 3, 4, 5, 6, 7]$ y $[4, 7, 0, 5, 2, 6, 3, 1]$, se tiene la tabla de extremos

Elemento	Extremos	Elemento	Extremos
0	1, 7, 7, 5	4	3, 5, 1, 7
1	0, 2, 3, 4	5	4, 6, 0, 2
2	1, 3, 5, 6	6	5, 7, 2, 3
3	2, 4, 6, 1	7	6, 0, 4, 0

Mutación

En la mutación por intercambio, dos posiciones en el *array* son seleccionadas aleatoriamente y los valores de sus elementos son intercambiados.



Remplazo

Después de esto, los individuos identificados como los peores en la población actual se reemplazan por los nuevos individuos generados en el paso anterior. Esta operación ayuda a mantener la diversidad genética y mejora la calidad promedio de la población con el tiempo.

3. Resultados

Después de ejecutar el código se obtienen distintas soluciones al problema. La Figura 3 presenta la evolución del fitness promedio a medida que avanzan las generaciones en el abordaje del problema de las 8 reinas utilizando algoritmos genéticos. La gráfica ilustra claramente cómo el valor de fitness mejora progresivamente con cada generación.

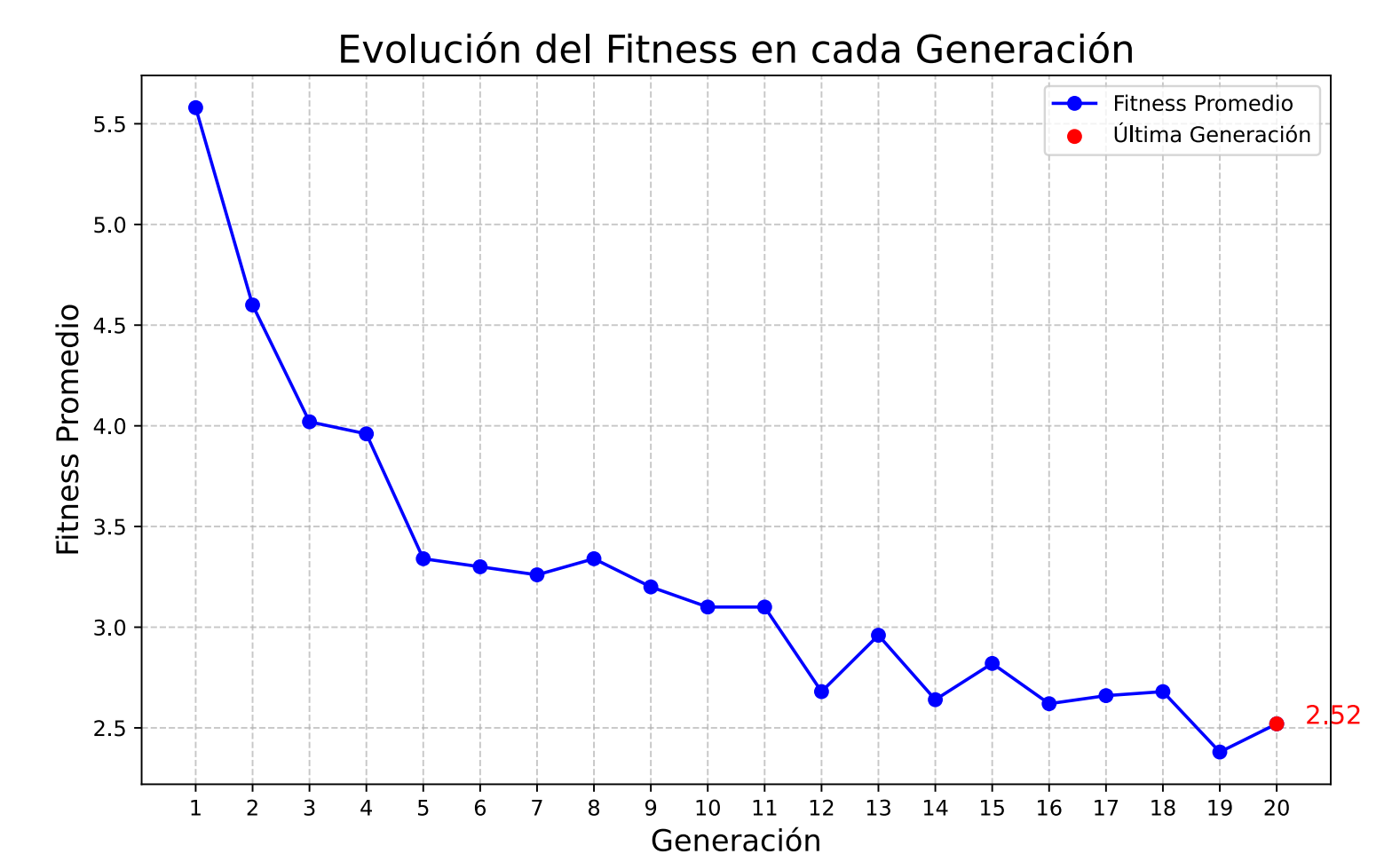


Figura 3: Promedio de fitness por generación

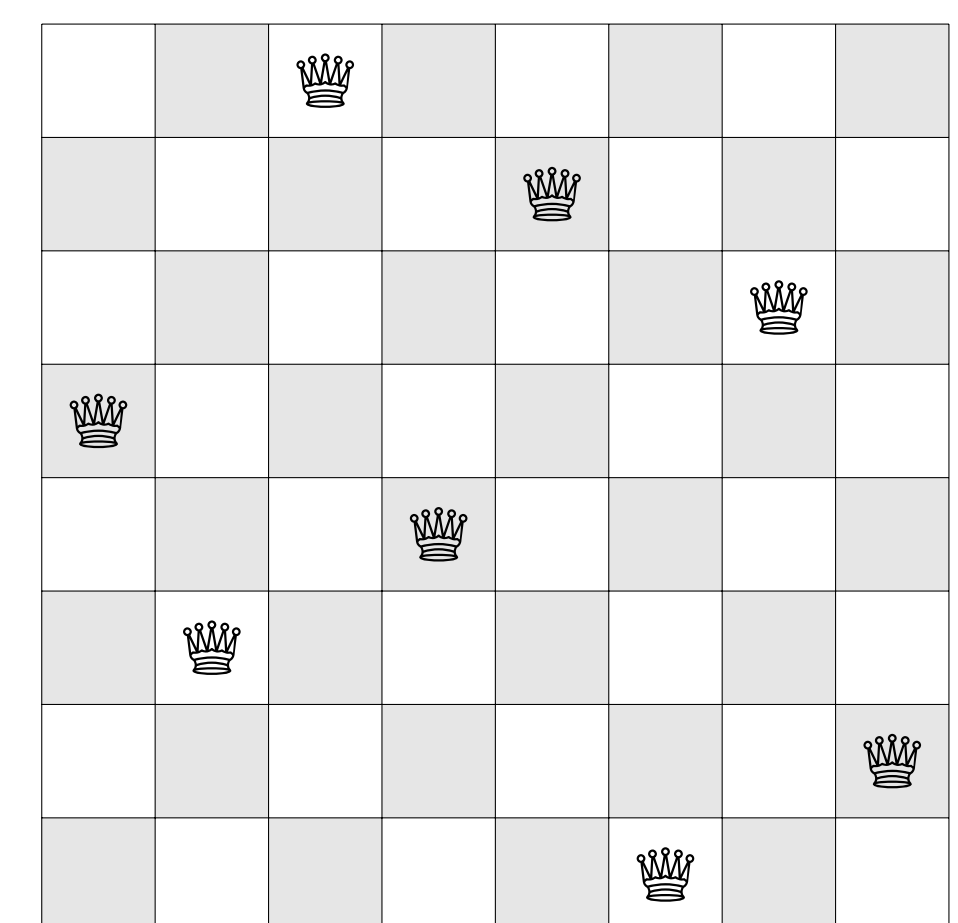


Figura 4: Ejemplo de solución al problema de las 8 reinas.

4. Conclusiones

En este estudio, hemos aplicado con éxito algoritmos genéticos para abordar el problema de las 8 reinas. La evolución del fitness a lo largo de las generaciones muestra una mejora constante, respaldando la eficacia de nuestro enfoque. Los resultados obtenidos subrayan la capacidad de los algoritmos genéticos para encontrar soluciones óptimas en problemas combinatorios.

La aplicación específica de representación de permutación, inicialización aleatoria, selección de padres universal estocástica, cruce a los extremos, mutación mezcla y reemplazo de brecha generacional ha demostrado ser exitosa en la resolución del problema de las 8 reinas.

Referencias

- [1] Eiben, A. E., & Smith, J. (2015). *Introduction to Evolutionary Computing*. En *Natural Computing Series*. Springer-Verlag Berlin Heidelberg.
- [2] Russell, S. J., & Norvig, P. (2004). *Inteligencia artificial: un enfoque moderno*. Prentice Hall.
- [3] Wirsansky, E. (2020). *Hands-On Genetic Algorithms with Python*. Packt Publishing eBooks.