1. Algoritmo genético simple

a) Escribe la modelación de las posibles soluciones:

Se usó una cadena de binarios que representa el genotipo, la longitud de la cadena es de 18 dígitos ya que con la conversión utilizada (binario a flotante entre 0 y 1) se necesita de un decimal de 9 dígitos para que nuestro fenotipo pueda tener una presición de 6 decimales y al necesitar una coordenada x y una y se necesita $2 * 9$ bits. El proceso usado para convertir de genotipo a fenotipo es el siguiente:

1. Dividir la lista binaria en partes iguales.
2. Pasar cada lista binaria $[b_1, ..., b_9]$ a la función `bit_string_to_float` $= \sum_{i \in 1...9} b_i * 10^{-i}$
3. Normalizar el valor regresado en el paso anterior a la función `normaliza_fenotipo`, que convierte el número flotante en un número entre el rango $[-500, 500]$

b) Establecer una población fija de 100 individuos, la tasa de recombinación en $P_c = 0.5$, la tasa de mutación en $P_m = 0.1$, la condición de término en 500 generaciones. ¿Cuál fue el mejor individuo al terminar la ejecución? Reporte la estructura del genotipo, fenotipo y fitness ¿es el máximo global?

El mejor individuo generado en una ejecución aleatoria fue:

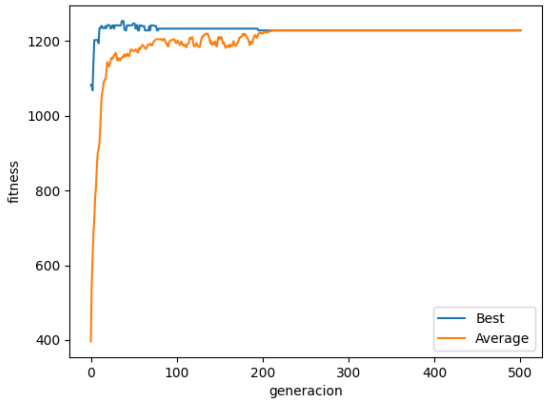
a) Genotipo: $[1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1]$

b) Fenotipo: $[416.015625, 431.640625]$

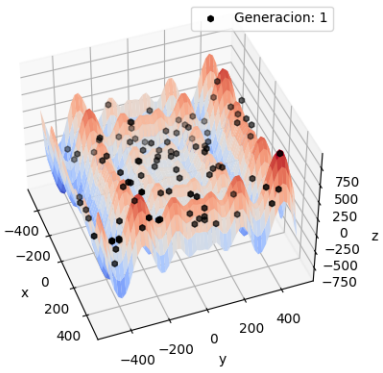
c) Schwefel(Fenotipo): 820.5315588989117

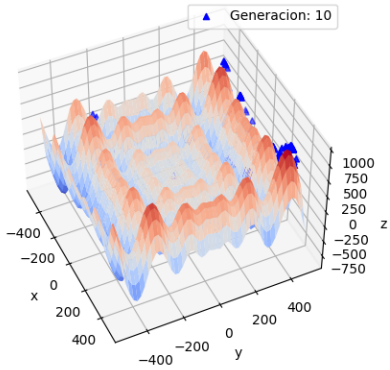
No es el máximo global pero está muy cerca. Las gráfica obtenidas fueron:

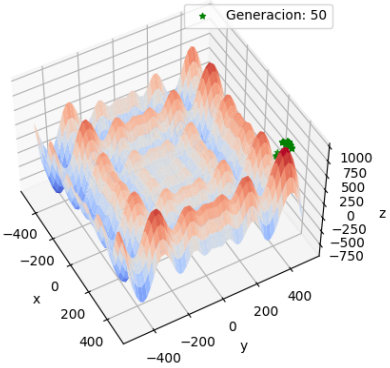
1. Generación vs Fitness:

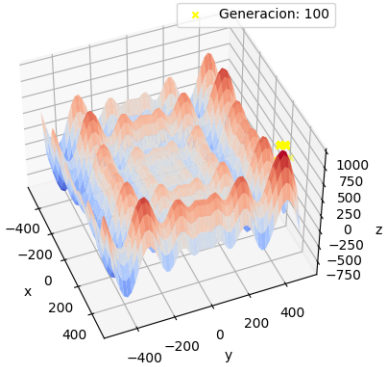


2. Distintas generaciones:



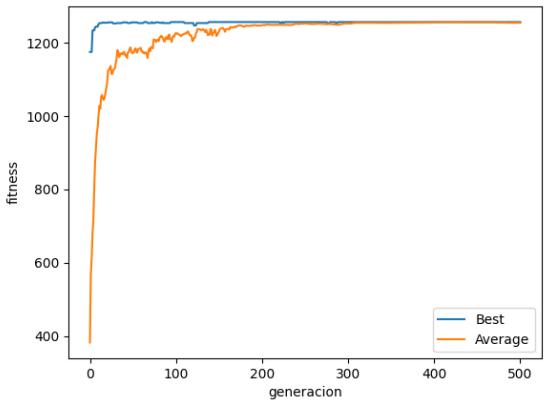




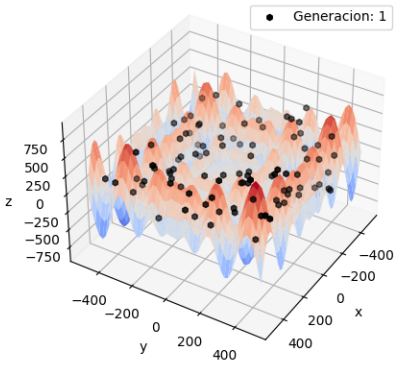


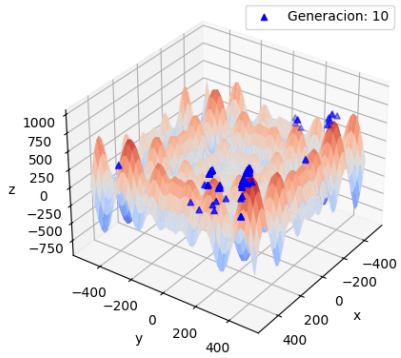
c) Genere otros experimentos cambiando la tasa de recombinación y de mutación. A partir de su experiencia trate de ajustar los parámetros para que el AGS tenga el mejor desempeño posible y graficar como en los puntos anteriores. ¿Cuáles fueron los parámetros con los que encontraron rápidamente el máximo global? Explique.

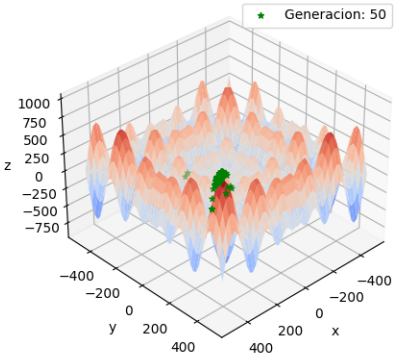
1. Generación vs Fitness:

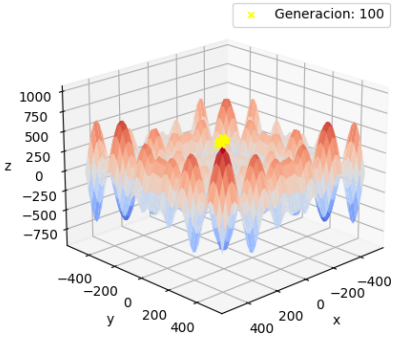


2. Distintas generaciones:







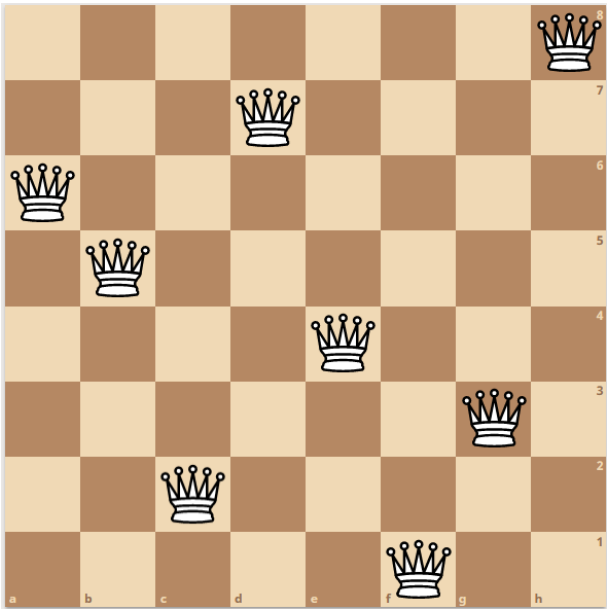


Se usaron los parámetros siguientes: $P_c = 0.5$ y $P_m = 0.01$. La taza de mutación se mantuvo en 0.5, ya que al ser el tamaño de la población relativamente grande se alcanzaba a generar muy buenos individuos casi desde las primeras generaciones. Para mantener estos individuos a lo largo de las generaciones solamente se hizo la cruza entre la mitad de los individuos seleccionados. Para la mutación se mantuvo lo más baja posible para que solamente en caso de que al inicio no tengamos individuos buenos puedan cambiar y adaptarse hasta obtener uno relativamente bueno.

2. El problema de las n reinas

a) Genotipo: Se usa el genotipo visto en clase: Una permutación de los números $1, ..., 8$, y el vector $g = (i1, ..., i8)$ denota una única configuración en el tablero donde la n-ésima columna, contiene exactamente una reina colocada en el i-ésimo renglón. Ej. [6,5,2,7,4,1,3,8]

b) Fenotipo: El tablero asociado al genotipo:

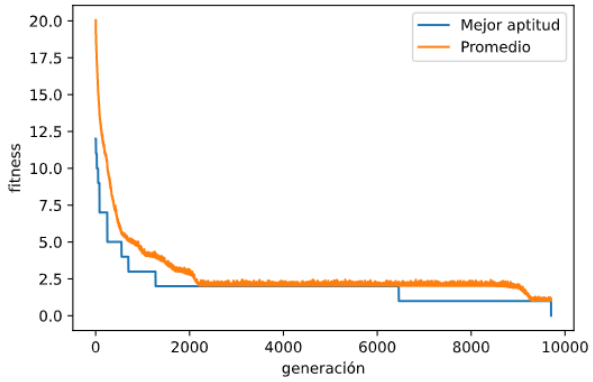


c) Fitness: Se usa la suma de amenzas en el tablero. Se busca minimizar el valor de fitness hasta llegar a 0. Fitness del ejemplo anterior = 4

A continuación se muestra una solución para el problema de las n reinas con $n = 30$:

[2, 12, 16, 11, 17, 23, 27, 10, 18, 28, 14, 8, 3, 30, 15, 24, 22, 29, 13, 4, 25, 1, 5, 9, 6, 19, 26, 20, 7, 21]

Generación vs fitness de la solución:



3. Hormiga artificial

a) Árbol óptimo encontrado:

```
to-report best-tree
 let best
  ["IF_FOOD_AHEAD"
    ["MOVE"]
    ["PROGN3"
      ["LEFT"]
      ["PROGN2"
        ["IF_FOOD_AHEAD" ["MOVE"] ["RIGHT"]]
        ["PROGN2"
          ["RIGHT"]
          ["PROGN2" ["LEFT"] ["RIGHT"]]]]
      ["PROGN2"
        ["IF_FOOD_AHEAD" ["MOVE"] ["LEFT"]]
        ["MOVE"]]]]
 report best
end
```

b) Gráfica generación vs fitness: