

Algoritmo Genético para resolver el problema de N-Reinas

Guillermo I. Bautista G. *Estudiante, UPIIZ*

Resumen—Cómo ya sabemos, hay muchas maneras de atacar un problema para resolverlo, sin importar la magnitud del mismo, si se comprende se puede construir, recrear y sobre todo solucionar en el caso de su verdadero entendimiento, los algoritmos genéticos intentan recrear los aspectos de la evolución de un problema, para que se auto analice y acabe resolviéndose solo

Index Terms—Álgoritmo Genético, N-Reinas

1. INTRODUCTION

LOS Algoritmos genéticos trabajan un conjunto de soluciones, o fenotipos como se les conoce, y un conjunto de individuos de una población, su solución suele ser única así que se describe y codifica en una cadena, generalmente binaria, considerada cromosoma y cada uno de sus datos genes, en el caso binario genotipo. Estos "evolucionan" por generaciones (1 o un conjunto de iteraciones), dándonos una mejor población y por ende un mejor resultado.

Esta mejora a través de generación se debe a operadores genéticos por iteración:

- selección
- cruza
- mutación
- reemplazo

Todos los operadores hacen referencia a algún factor importante en la evolución biológica como la selección que se refiere a una selección natural, eliminando a los menos aptos para ser una solución.

2. PROBLEMA DE N-REINAS

El problema consiste en encontrar una distribución de n reinas de ajedrez en un tablero

de $n * n$, de tal manera en que ninguna reina se pueda atacar, lo que nos dice que no puede haber 2 reinas en la misma columna o diagonal. Este problema puede interpretarse de 2 maneras:

- Encontrando una solución única
- Encontrando todas las soluciones posibles de un tablero $n * n$

Nosotros nos centraremos en encontrar 1 solución para distintos valores n con un algoritmo genético, modificando los distintos parametros dados

3. RESULTADOS

Para resolver un $n = 8$, tuvimos que utilizar los siguientes parámetros:

- 2000 generaciones
- un tamaño de población de 30
- 20 % de probabilidad de muta
- 0.1 % de probabilidad de reemplazo

la cruza esta definida cuando la selección elimina población Para resolver un $n = 15$, tuvimos

```

g: 301 t:2 id:4/4695494
g: 302 f:2 id:474695494
g: 303 f:2 id:474695494
g: 304 f:2 id:474695494
g: 305 f:2 id:474695494
g: 306 f:0 id:474695494
g: 306 [4, 7, 3, 0, 6, 1, 5, 2]
BUILD SUCCESSFUL (total time: 0 seconds)

```

Figura 1. N=8

```
g: 3636 f:2 id:474695494
g: 3637 f:2 id:474695494
g: 3638 f:2 id:474695494
g: 3639 f:2 id:474695494
g: 3640 f:2 id:474695494
g: 3641 f:0 id:474695494
g: 3641 [20, 4, 16, 27, 3, 12, 15, 9, 19, 14, 1, 21, 0, 29, 10, 7, 28, 23, 11, 17, 5, 25, 6, 18, 13, 8, 26, 24, 22, 2]
BUILD SUCCESSFUL (total time: 0 seconds)
```

Figura 2. N=30

que utilizar los siguientes parámetros:

- 8000 generaciones
- un tamaño de población de 35
- 20 % de probabilidad de muta
- 0.1 % de probabilidad de remplazo

```
g: 5203 f:2 id:474695494
g: 5204 f:2 id:474695494
g: 5205 f:2 id:474695494
g: 5206 f:2 id:474695494
g: 5207 f:2 id:474695494
g: 5208 f:0 id:474695494
g: 5208 [10, 3, 5, 11, 8, 6, 2, 12, 14, 0, 7, 4, 1, 13, 9]
BUILD SUCCESSFUL (total time: 0 seconds)
```

Figura 3. N=15

Para resolver un n = 30, tuvimos que utilizar los siguientes parámetros:

- 5000 generaciones
- un tamaño de población de 30
- 20 % de probabilidad de muta
- 0.1 % de probabilidad de remplazo

Dándonos cuenta que esta solución consume menos recursos que la de 15, resultados en la Figura 2

Para resolver un n = 70, tuvimos que utilizar los siguientes parámetros:

- 20000 generaciones
- un tamaño de población de 30
- 20 % de probabilidad de muta
- 0.1 % de probabilidad de remplazo

donde solo tuvimos que agregar generaciones para dar un resultado, en 4, con la respuesta g: 8094 [14, 3, 26, 34, 29, 64, 23, 39, 46, 57, 63, 10, 28, 48, 60, 44, 7, 18, 54, 49, 12, 31, 5, 20, 27, 9, 4, 35, 22, 50, 52, 8, 6, 67, 15, 42, 47, 55, 66, 37, 59, 30, 13, 1, 53, 25, 19, 0, 38, 11, 45, 56, 65, 32, 69, 2, 68, 61, 40, 16, 24, 33, 58, 17, 51, 41, 21, 36, 43, 62]

```
g: 8088 f:2 id:474695494
g: 8089 f:2 id:474695494
g: 8090 f:2 id:474695494
g: 8091 f:2 id:474695494
g: 8092 f:2 id:474695494
g: 8093 f:2 id:474695494
g: 8094 f:0 id:474695494
g: 8094 [14, 3, 26, 34, 29, 64, 23, 39, 46, 57, 63, 10, 28, 48, 60, 44, 7, 18, 54, 49, 12, 31, 5, 20, 27, 9, 4, 35, 22, 50, 52, 8, 6, 67, 15, 42, 47, 55, 66, 37, 59, 30, 13, 1, 53, 25, 19, 0, 38, 11, 45, 56, 65, 32, 69, 2, 68, 61, 40, 16, 24, 33, 58, 17, 51, 41, 21, 36, 43, 62]
BUILD SUCCESSFUL (total time: 4 seconds)
```

Figura 4. N=70

Para resolver un n = 90, tuvimos que utilizar los siguientes parámetros:

- 20000 generaciones
- un tamaño de población de 35
- 25 % de probabilidad de muta
- 1.5 % de probabilidad de remplazo

con la respuesta g: 8276 [59, 18, 27, 46, 39, 81, 64, 76, 74, 31, 28, 19, 9, 26, 20, 83, 45, 67, 29, 52, 13, 53, 38, 8, 55, 23, 62, 75, 68, 66, 85, 33, 77, 56, 58, 88, 40, 7, 3, 14, 86, 32, 24, 63, 12, 89, 34, 21, 4, 1, 80, 78, 61, 25, 73, 2, 37, 44, 79, 51, 72, 60, 0, 42, 5, 43, 11, 57, 71, 84, 50, 10, 15, 87, 69, 41, 35, 54, 47, 6, 17, 65, 22, 36, 70, 48, 82, 16, 30, 49]

```
g: 8270 f:2 id:474695494
g: 8271 f:2 id:474695494
g: 8272 f:2 id:474695494
g: 8273 f:2 id:474695494
g: 8274 f:2 id:474695494
g: 8275 f:2 id:474695494
g: 8276 f:0 id:474695494
g: 8276 [59, 18, 27, 46, 39, 81, 64, 76, 74, 31, 28, 19, 9, 26, 20, 83, 45, 67, 29, 52, 13, 53, 38, 8, 55, 23, 62, 75, 68, 66, 85, 33, 77, 56, 58, 88, 40, 7, 3, 14, 86, 32, 24, 63, 12, 89, 34, 21, 4, 1, 80, 78, 61, 25, 73, 2, 37, 44, 79, 51, 72, 60, 0, 42, 5, 43, 11, 57, 71, 84, 50, 10, 15, 87, 69, 41, 35, 54, 47, 6, 17, 65, 22, 36, 70, 48, 82, 16, 30, 49]
BUILD SUCCESSFUL (total time: 7 seconds)
```

Figura 5. N=90

4. CONCLUSIONES

Los algoritmos genéticos son un tipo de algoritmos útiles a la hora de necesitar crear un proceso repetitivo donde en cada iteración se reevalúa la entrada para cada vez pulir más un resultado, estos no siempre terminan en éxitos, en la mayoría de las ocasiones terminan las generaciones establecidas sin ningún resultado y eso está bien porque se hacen de esa manera, un algoritmo de optimización se puede llegar a estancar con parámetros establecidos y sin embargo esta optimización puede ser algo lejano a un resultado.