



INSTITUTO POLITÉCNICO NACIONAL



Análisis de Algoritmos

Algoritmos Genéticos

Docente: Roberto Oswaldo Cruz Leija

Alumno: Héctor Mauricio Zamudio Domínguez

Grupo: 3CM1

Fecha: 29/11/2019

Introducción

Los algoritmos basados en los principios de la evolución natural se utilizan en problemas donde no se pueden encontrar soluciones de manera sencilla, su funcionamiento general es generar de forma aleatoria una población inicial de soluciones potenciales, se entra en un proceso iterativo que transforma la población a través de una evaluación de las soluciones que forman la población, selecciona los “mejores” y la reproducción de estos formando una nueva población.

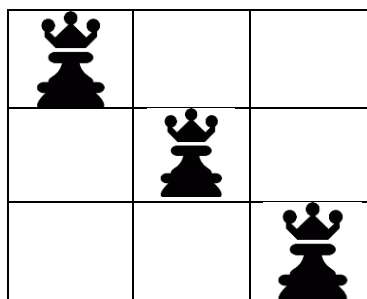
“El problema de las n -Reinas(n -Queens Problem) es muy antiguo, propuesto por primera vez en el año de 1848, consiste en encontrar una asignación a n reinas en un tablero de ajedrez de $n \times n$ de modo tal, que estas no se ataquen.”

Se implemento un algoritmo genético que da soluciones al problema de n -Reinas, ya que es un problema complicado de resolver y se requieren estrategias avanzadas para resolverlos.

Desarrollo

En el caso del desarrollo del programa, es importante hacer una reflexión en cuanto a la dificultad del modelado del problema, ya que aun que el problema de las n -Reinas, no tiene demasiadas restricciones es significativo elegir una buena estructura de datos que ayuda a mejorar el rendimiento del algoritmo. La estructura implementada en el programa es el siguiente.

	Posición x1	Posición x2	Posición x3
Posición y	0	1	2



La estructura mostrada anteriormente describe la figura mostrada. La construcción de la estructura no permite tener dos reinas en una misma columna.

Se crean diferentes configuraciones las cuales recibe como parámetros

- el número de generación: Cuantas veces va a iterar el programa si no encuentra una solución),
- el tamaño de la población,
- una probabilidad de muta para evitar estancamiento en la población, generalmente se utiliza una probabilidad baja,
- Probabilidad de cruce: Probabilidad de cruzar dos miembros de la población
- N-Reinas: Buscar una solución para un tablero de nxn
- Fitness: Es el parámetro con el cual decidimos cual gen reproducir, para este caso, nuestro fitness es el número de ataques total entre las reinas del tablero y la meta es reducirlo a cero(Encontrar una solución).

Resultados

Numero de generaciones	50000	g: 259 f:2 id:1911972153 g: 260 f:2 id:1911972153 g: 261 f:2 id:1911972153 g: 262 f:2 id:1911972153 g: 263 f:2 id:1911972153 g: 264 f:2 id:1911972153 g: 265 f:2 id:1911972153 g: 266 f:2 id:1911972153 g: 267 f:2 id:1911972153 g: 268 f:0 id:1911972153 g: 268 [5, 2, 4, 6, 0, 3, 1, 7]
Tamaño población	20	
Probabilidad de Mutación	0.2	
Probabilidad de Muestra	0.001	
Reinas	8	

Numero de generaciones	50000	g: 259 f:2 id:1911972153 g: 260 f:2 id:1911972153 g: 261 f:2 id:1911972153 g: 262 f:2 id:1911972153 g: 263 f:2 id:1911972153 g: 264 f:2 id:1911972153 g: 265 f:2 id:1911972153 g: 266 f:2 id:1911972153 g: 267 f:2 id:1911972153 g: 268 f:0 id:1911972153 g: 268 [5, 2, 4, 6, 0, 3, 1, 7]
Tamaño población	20	
Probabilidad de Mutación	0.2	
Probabilidad de Muestra	0.001	
Reinas	8	

Numero de generaciones	50000	
------------------------	-------	--

Tamaño población	20	g: 149 f:2 id:1470333138 g: 150 f:2 id:1470333138 g: 151 f:2 id:1470333138 g: 152 f:2 id:1470333138 g: 153 f:2 id:1470333138 g: 154 f:2 id:1470333138 g: 155 f:2 id:1470333138 g: 156 f:2 id:1470333138 g: 157 f:2 id:1470333138 g: 158 f:0 id:1470333138 g: 158 [9, 1, 12, 5, 11, 6, 0, 10, 4, 14, 8, 2, 7, 3, 13]
Probabilidad de Mutación	0.2	
Probabilidad de Muestra	0.001	
Reinas	15	

Numero de generaciones	50000
Tamaño población	20
Probabilidad de Mutación	0.2
Probabilidad de Muestra	0.001
Reinas	30

g: 1632 f:2 id:1470333138
g: 1633 f:2 id:1470333138
g: 1634 f:2 id:1470333138
g: 1635 f:2 id:1470333138
g: 1636 f:2 id:1470333138
g: 1637 f:2 id:1470333138
g: 1638 f:2 id:1470333138
g: 1639 f:2 id:1470333138
g: 1640 f:2 id:1470333138
g: 1641 f:0 id:1470333138
g: 1641 [13, 4, 20, 8, 5, 19, 2, 23, 7, 26, 29, 21, 16, 25, 12, 6, 24, 0, 11, 1, 22, 15, 28, 10, 3, 9, 18, 27, 17, 14]

Numero de generaciones	50000
Tamaño población	20
Probabilidad de Mutación	0.2
Probabilidad de Muestra	0.001
Reinas	70

g: 47210 f:2 id:1470333138
g: 47211 f:2 id:1470333138
g: 47212 f:2 id:1470333138
g: 47213 f:2 id:1470333138
g: 47214 f:2 id:1470333138
g: 47215 f:2 id:1470333138
g: 47216 f:0 id:1470333138
g: 47216 [24, 53, 7, 22, 25, 32, 64, 8, 26, 5, 38, 52, 60, 36, 12, 51, 45, 67, 34, 19, 42, 35, 37, 55, 17, 10, 61, 0, 16, 1, 43, 57, 49, 13, 68, 66, 44, 27, 15, 28, 2, 6, 23, 46, 39, 29, 14, 3, 9, 56, 54, 62, 58, 47, 21, 31, 65, 69, 40, 18, 59, 30, 20, 33, 50, 4, 63, 41, 11, 48]

Numero de generaciones	500000 *
Tamaño población	20
Probabilidad de Mutación	0.2
Probabilidad de Muestra	0.001
Reinas	90

g: 14907 f:2 id:1470333138
g: 14908 f:2 id:1470333138
g: 14909 f:2 id:1470333138
g: 14910 f:2 id:1470333138
g: 14911 f:2 id:1470333138
g: 14912 f:2 id:1470333138
g: 14913 f:0 id:1470333138
g: 14913 [43, 56, 44, 16, 58, 83, 9, 68, 24, 1, 45, 41, 29, 76, 2, 89, 14, 22, 32, 30, 53, 65, 6, 38, 46, 51, 12, 50, 66, 80, 67, 37, 13, 78, 86, 88, 23, 34, 79, 8, 61, 77, 18, 71, 69, 35, 20, 7, 10, 59, 84, 21, 17, 3, 62, 26, 74, 48, 5, 31, 39, 4, 25, 64, 19, 47, 75, 15, 87, 33, 36, 27, 72, 85, 0, 60, 49, 55, 73, 54, 11, 40, 28, 82, 42, 52, 63, 70, 81, 57]

Conclusiones

El programa realizado necesito técnicas de programación de niveles mas avanzados, pero esto es debido a que el problema de las n-Reinas necesitaba un programa así de potente.

Aunque la configuración básicamente es la misma para todos los casos, en número de la generación en la cual se resolvía el problema también iba incrementando, y para el ultimo caso fue necesario incrementar el tamaño de la población al orden de los millones de pobladores para encontrar una solución.

Referencias

<https://www.cs.buap.mx/~zacarias/FZF/nreinas3.pdf>