



PROYECTO N-REINAS

Por

ANDRÉS HUMBERTO AGREDO BERMUDEZ

CHRISTIAN ANDRES MORALES MERA

JEFFERSON AMADO PEÑA TORRES

Dirigido a

ANGEL GARCIA BAÑOS

Asignatura

Computación Evolutiva

Facultad de Ingeniería

Escuela de Ingeniería de Sistemas y Computación

Programa Académico de Ingeniería de Sistemas

Santiago de Cali, Junio de 2015

Diversidad de la aptitud

Para determinar la actitud se fijó mediante los ataques de las reinas es decir:

El cromosoma elimina los conflictos de fila y columna, las soluciones equivocadas solo tienen ataques diagonales entre reinas. El método de la figura 1 nos cuenta el número de ataques diagonales entre las reinas y devuelve un valor de fitness. Los valores de fitness deben contar ataques diagonales.

```
def count_attacks()
  #inicialmente la actitud es 0
  @fitness = 0
  #para recorrer las diagonales
  (0..@n_queens-2).each do |i|
    ((i+1)..@n_queens-1).each do |j|
      if i + self[i] == j + self[j] or i - self[i] == j - self[j]
        #el incremento es entero, buscamos minimizar
        @fitness += 1
      end
    end
  end
  #retorno la actitud
  @fitness
end
```

Figura 1: Algoritmo que calcula número de ataques

El siguiente paso una vez determinado el número de ataques, es determinar la mutación del cromosoma el cual consta del algoritmo de la figura 2 el cual nos dice que a partir de un cromosoma padre el genera aleatoriamente los siguientes cromosomas hijos y al nuevo cromosoma se le determina la actitud como anteriormente se explicó.

```
def mutate()
  #Se realiza una copia profunda del cromosoma
  son = self.clone
  #Se obtienen las posiciones a cambiar aleatoriamente
  source = rand(@n_queens)
  target = rand(@n_queens)
  #Si las posiciones son diferentes
  if source != target
    tmp = son[source]
    son[source] = son[target]
    son[target] = tmp
  #Si las posiciones son iguales
  elsif source == target
    #si es menor que el limite le aumento 1 al destino
    if target < (@n_queens-1)
      target+=1;
    #Si es menor se lo resto
    else
      target-=1;
    end
    #hago el cambio
    tmp = son[source]
    son[source] = son[target]
    son[target] = tmp
  end
  #Cuento nuevamente los ataques
  son.count_attacks
  #Retorno el hijo resultante
  son
end
```

Figura 2: Algoritmo que genera las mutaciones

Algoritmo genético

El algoritmo genético para el caso de las n reinas funciona de la siguiente manera:

Cada iteración del algoritmo se genera un número de cromosomas el cual genera un nuevo individuo el cual sería el hijo.

Este hijo es generado por una mutación y se reemplazan la peor parte del ranking de la población.

Para una solución correcta, la función de aptitud regresará cero.

Resultados

Los siguientes resultados representan la diversidad de la selección.

La siguiente tabla representa los resultados para 10 repeticiones y un máximo de generaciones de 1000 y 2000 mutaciones

| N de reinas | Diversidad |
|-------------|------------|
| 8 | 7 |
| 10 | 9 |
| 30 | 29 |
| 50 | 49 |
| 75 | 74 |
| 100 | 99 |
| 200 | 199 |
| 300 | 299 |
| 500 | 499 |

Tabla 1: primeros resultados obtenidos en la aplicación.

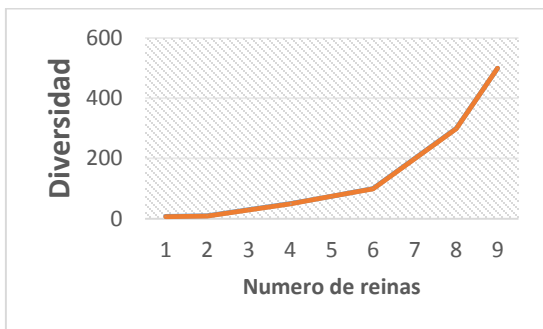


Figura 3: resultados obtenidos al ejecutar la aplicación con los datos suministrados y obtenidos en la tabla 1.

La siguiente tabla muestra los resultados variando la entre 10000 y 100000 el número de generaciones dejando igual el número de mutaciones.

```
def initialize(num_queens = 8, num_chromosomes = 2000, maxnum_generations = 10000, debug = false)

C:\Windows\system32\cmd.exe

c:\Users\hjp\Documents\ejemplos>ruby n-queens.rb
D: ?
D: ?
> Solution: Chromosome: [2, 4, 7, 3, 0, 6, 1, 5] with fitness 0
c:\Users\hjp\Documents\ejemplos>
```

Figura 4: resultados obtenidos para un numero de cromosomas de 8, 2000 mutaciones y 10000 generaciones.

| N de reinas | Diversidad |
|-------------|------------|
| 8 | 7 |
| 10 | 9 |
| 30 | 29 |
| 50 | 49 |
| 75 | 74 |
| 100 | 99 |
| 200 | 199 |
| 300 | 299 |
| 500 | 499 |

Tabla 2: resultados variando el número de generaciones.

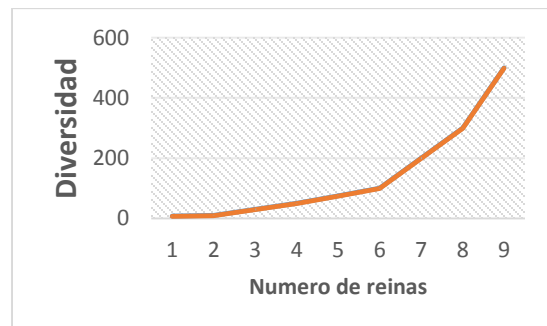


Figura 5: resultados obtenidos de al ejecutar la aplicación con los datos suministrados y obtenidos de la tabla 2.

En el grafico se observa el número de reinas vs diversidad del algoritmo genético.

En caso de la tabla 2 se varía en el número de generación entre 10000 y 100000 como se aprecia en la figura 4 que se especifica 8 reinas y 10000 generaciones dando el mismo resultado obtenido en la tabla 1 diferenciando el tiempo de ejecución del algoritmo ya que podemos observar una espera sustancial con respecto al primero.

Además el comportamiento de los gráficos para estos casos es el esperado ya que debería ser en tiempo ascendente.

Y siempre que obtengamos una aptitud igual a cero podemos determinar la correctitud del algoritmo

A continuación se muestra los resultados obtenidos para el caso de ocho (8) reinas y sus correspondientes mutaciones.

```
0 : Chromosome: [4, 6, 3, 0, 2, 7, 5, 1] with fitness 0
1 : Chromosome: [0, 6, 1, 5, 2, 4, 7, 3] with fitness 1
2 : Chromosome: [1, 3, 6, 4, 7, 5, 0, 2] with fitness 1
3 : Chromosome: [5, 3, 0, 7, 4, 6, 1, 2] with fitness 1
4 : Chromosome: [2, 7, 5, 1, 3, 0, 6, 4] with fitness 1
5 : Chromosome: [0, 2, 4, 7, 5, 3, 1, 6] with fitness 1
6 : Chromosome: [5, 4, 6, 1, 3, 7, 0, 2] with fitness 1
7 : Chromosome: [6, 4, 1, 7, 5, 3, 2, 0] with fitness 1
8 : Chromosome: [2, 4, 6, 0, 3, 1, 5, 7] with fitness 1
9 : Chromosome: [4, 7, 0, 3, 6, 2, 5, 1] with fitness 1
10 : Chromosome: [3, 5, 2, 1, 6, 0, 7, 4] with fitness 1
11 : Chromosome: [4, 0, 5, 2, 6, 1, 3, 7] with fitness 1
12 : Chromosome: [7, 4, 1, 3, 6, 0, 2, 5] with fitness 1
13 : Chromosome: [0, 5, 7, 2, 4, 6, 1, 3] with fitness 1
14 : Chromosome: [2, 6, 5, 1, 4, 0, 7, 3] with fitness 1
15 : Chromosome: [7, 2, 6, 1, 4, 0, 5, 3] with fitness 1
16 : Chromosome: [2, 7, 3, 0, 1, 4, 6, 5] with fitness 1
17 : Chromosome: [0, 6, 3, 7, 2, 4, 5, 1] with fitness 1
18 : Chromosome: [3, 5, 2, 4, 6, 0, 7, 1] with fitness 1
19 : Chromosome: [5, 2, 0, 6, 3, 7, 1, 4] with fitness 1
20 : Chromosome: [2, 0, 5, 7, 4, 1, 3, 6] with fitness 1
21 : Chromosome: [2, 5, 7, 4, 1, 0, 6, 3] with fitness 1
22 : Chromosome: [3, 0, 7, 5, 2, 6, 1, 4] with fitness 1
23 : Chromosome: [5, 0, 2, 4, 6, 7, 3, 1] with fitness 1
24 : Chromosome: [1, 3, 4, 0, 7, 5, 2, 6] with fitness 1
25 : Chromosome: [4, 6, 1, 3, 5, 7, 2, 0] with fitness 1
26 : Chromosome: [6, 3, 0, 4, 1, 7, 5, 2] with fitness 1
27 : Chromosome: [5, 4, 1, 7, 2, 6, 3, 0] with fitness 1
28 : Chromosome: [4, 0, 5, 3, 1, 6, 7, 2] with fitness 1
29 : Chromosome: [2, 7, 3, 6, 0, 4, 1, 5] with fitness 1
30 : Chromosome: [7, 4, 2, 0, 5, 3, 1, 6] with fitness 1
31 : Chromosome: [6, 3, 1, 7, 4, 0, 2, 5] with fitness 1
32 : Chromosome: [5, 1, 4, 7, 0, 6, 3, 2] with fitness 1
33 : Chromosome: [5, 1, 4, 6, 0, 2, 7, 3] with fitness 1
34 : Chromosome: [6, 3, 0, 4, 1, 5, 7, 2] with fitness 1
35 : Chromosome: [0, 4, 6, 5, 3, 1, 7, 2] with fitness 1
36 : Chromosome: [4, 6, 1, 5, 7, 3, 0, 2] with fitness 1
37 : Chromosome: [3, 4, 0, 7, 5, 2, 6, 1] with fitness 1
38 : Chromosome: [1, 7, 5, 2, 6, 3, 0, 4] with fitness 1
39 : Chromosome: [5, 1, 4, 7, 3, 6, 0, 2] with fitness 1
40 : Chromosome: [1, 7, 4, 2, 0, 6, 3, 5] with fitness 1
41 : Chromosome: [1, 5, 3, 0, 6, 4, 2, 7] with fitness 1
42 : Chromosome: [1, 6, 3, 0, 7, 4, 2, 5] with fitness 1
43 : Chromosome: [1, 5, 2, 6, 3, 7, 0, 4] with fitness 1
44 : Chromosome: [6, 1, 7, 4, 3, 0, 2, 5] with fitness 1
45 : Chromosome: [5, 2, 0, 4, 7, 1, 3, 6] with fitness 1
46 : Chromosome: [2, 6, 1, 3, 0, 4, 7, 5] with fitness 1
```

Figura 6: resultados obtenidos para 8 reinas y una población de 2000

En este caso lo que busca el programa es determinar la aptitud igual a cero y se saca

como solución el proceso se realiza con varias reinas y siempre buscando ese objetivo final.

Variabilidad con respecto al fitness

El siguiente experimento consiste en graficar para un determinado número de reinas su comportamiento con respecto a la aptitud y el número de mutaciones para cada una de estas.

A continuación se puede observar el comportamiento con un total de 2000 mutaciones para ambos casos

| N de reinas | fitness | mutación |
|-------------|---------|----------|
| 8 | 0 | 1 |
| 8 | 1 | 56 |
| 8 | 2 | 132 |
| 8 | 3 | 336 |
| 8 | 4 | 378 |
| 8 | 5 | 386 |
| 8 | 6 | 322 |
| 8 | 7 | 159 |
| 8 | 8 | 122 |
| 8 | 9 | 36 |
| 8 | 10 | 26 |
| 8 | 11 | 22 |
| 8 | 12 | 6 |
| 8 | 13 | 4 |
| 8 | 14 | 4 |
| 8 | 15 | 1 |
| 8 | 16 | 7 |
| 8 | 8 | 1 |
| 8 | 5 | 1 |

Tabla 4: resultados para 8 reinas y el comportamiento de la aptitud a lo largo de las 2000 mutaciones.

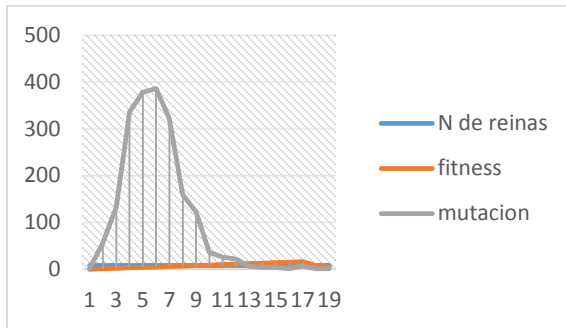


Figura 7: grafico del comportamiento de los datos de las mutaciones y numero de reinas.

Resultados obtenidos mediante la actitud son los representados en la tabla 3

| N de reinas | fitness | mutación |
|-------------|---------|----------|
| 12 | 0 | 7 |
| 12 | 1 | 113 |
| 12 | 2 | 458 |
| 12 | 3 | 1120 |
| 12 | 4 | 331 |
| 12 | 2 | 1 |

Tabla 3: resultados obtenidos variando la aptitud

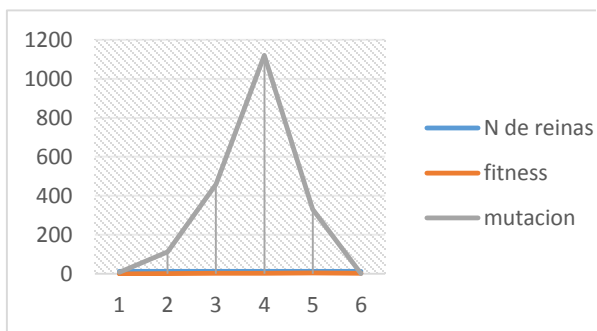


Figura 8: resultados obtenidos mediante la actitud y número de mutaciones con respecto al número de reinas.

Lo que podemos observar en la figura 7 y 8 es el número de mutaciones con respecto al fitness y numero de reinas. Se observa es un comportamiento de campana para ambos casos y siempre va llegar a un tope de mutaciones máxima y empieza a reducir debido a que empieza a descartar soluciones erróneas.

Variabilidad mixta

Para determinar este proceso se toma en cuenta ambos procesos es decir la diversidad y el fitness para determinar la variabilidad mixta, se calcula la suma de ambos casos y se divide por el número de generaciones un ejemplos de este caso se observa en la Tabla 4 y toma como solución el menos de estos casos.

| n de reinas | diversidad | fitness | mixto |
|-------------|------------|---------|--------------------------|
| A | 7 | 1 | 8/numero de generaciones |
| B | 5 | 3 | 8/numero de generaciones |
| C | 4 | 4 | 8/numero de generaciones |
| D | 6 | 2 | 8/numero de generaciones |
| E | 1 | 3 | 4/numero de generaciones |

Tabla 4: representación del cálculo mixto

Resultados obtenidos para la variabilidad mixta

| N de reinas | fitness | diversity | mixed | mutaciones |
|-------------|---------|-----------|-------|------------|
| 13 | 1 | 3 | 0 | 3 |
| 13 | 2 | 35 | 0 | 35 |
| 13 | 3 | 149 | 1 | 149 |
| 13 | 4 | 432 | 1 | 432 |
| 13 | 5 | 839 | 1 | 839 |
| 13 | 6 | 1313 | 2 | 1312 |
| 13 | 7 | 1492 | 2 | 1492 |
| 13 | 8 | 1500 | 2 | 1499 |
| 13 | 9 | 1258 | 3 | 1258 |
| 13 | 10 | 978 | 3 | 978 |
| 13 | 11 | 725 | 3 | 725 |
| 13 | 12 | 467 | 4 | 467 |
| 13 | 13 | 313 | 4 | 313 |
| 13 | 14 | 187 | 4 | 187 |
| 13 | 15 | 113 | 5 | 113 |
| 13 | 16 | 85 | 5 | 85 |
| 13 | 17 | 40 | 5 | 40 |
| 13 | 18 | 26 | 6 | 26 |
| 13 | 19 | 17 | 5 | 17 |
| 13 | 20 | 12 | 4 | 20 |
| 13 | 21 | 9 | 3 | 21 |
| 13 | 22 | 5 | 1 | 22 |
| 13 | 23 | 2 | 0 | 23 |
| 13 | 8 | 1500 | 2 | 8 |
| 13 | 6 | 1313 | 2 | 6 |

Tabla 5: resultados obtenidos para la representación mixta

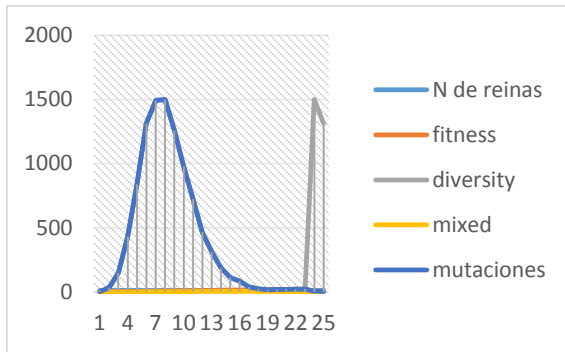


Figura 9: representación del número de reinas (13) respecto a su aptitud, diversidad y la variedad mixta

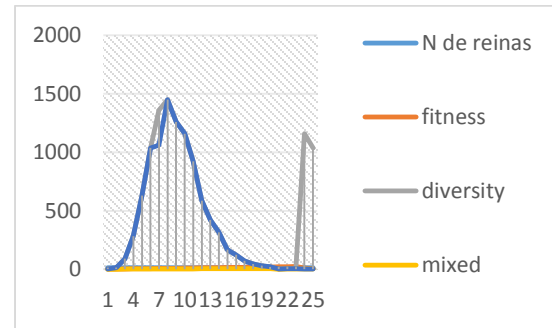


Figura 10: representación del número de reinas (14) respecto a su aptitud, diversidad y la variabilidad mixta

Segundo experimento para la variabilidad mixta se realiza teniendo en cuenta 14 reinas y 10000 individuos.

| N de reinas | fitness | diversity | mixed | mutaciones |
|-------------|---------|-----------|-------|------------|
| 14 | 1 | 1 | 0 | 1 |
| 14 | 2 | 12 | 0 | 12 |
| 14 | 3 | 91 | 1 | 91 |
| 14 | 4 | 283 | 1 | 283 |
| 14 | 5 | 630 | 1 | 630 |
| 14 | 6 | 1037 | 2 | 1036 |
| 14 | 7 | 1360 | 2 | 1060 |
| 14 | 8 | 1450 | 2 | 1450 |
| 14 | 9 | 1260 | 3 | 1260 |
| 14 | 10 | 1159 | 3 | 1158 |
| 14 | 11 | 924 | 3 | 924 |
| 14 | 12 | 587 | 4 | 587 |
| 14 | 13 | 421 | 4 | 421 |
| 14 | 14 | 315 | 4 | 315 |
| 14 | 15 | 167 | 5 | 167 |
| 14 | 16 | 123 | 5 | 123 |
| 14 | 17 | 72 | 5 | 72 |
| 14 | 18 | 45 | 6 | 45 |
| 14 | 19 | 31 | 6 | 31 |
| 14 | 20 | 21 | 6 | 21 |
| 14 | 21 | 2 | 0 | 2 |
| 14 | 22 | 5 | 1 | 5 |
| 14 | 23 | 4 | 1 | 4 |
| 14 | 10 | 1159 | 3 | 1 |
| 14 | 6 | 1037 | 2 | 1 |

Tabla 6: resultados obtenidos para un total de 14 reinas y una población de 10000 individuos.

En ambos casos se puede interpretar un mismo conocimiento cuando se genera en ese pico en la gráfica es debido a que busca nuevas poblaciones pero como encontró una solución válida el algoritmo da por terminado la búsqueda.

Conclusión

- Una vez realizado los experimentos se opta por una operación que garantice que la variabilidad mixta sea mejor que las demás ya que si se considerara una operación ya sea multiplicativo o exponencial siempre se verá un resultado peor a los anteriores.
- Cuando se realiza la operación mixta se observa que no encuentro una solución cuando tiene una población pequeña por eso se trabaja con valores grandes.

Referencias

(<http://www.zemris.fer.hr/~golub/clanci/iti2007.pdf>, 2015)
 (<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.7571&rep=rep1&type=pdf>, 2015)