UNIVERSIDAD MARIANO GALVEZ DE GUATEMALA
CENTRO UNIVERSITARIO DE MAZATENANGO - SUCHITEPEQUEZ
FACULTAD DE INGENIERIA EN SISTEMAS
INTELIGENCIA ARTIFICIAL - Ing. Miguel Lemus

Proyecto Algoritmo Genético

Introducción

Proyecto para el curso de inteligencia artificial como parte del aprendizaje en el funcionamiento de algoritmos genéticos los cuales buscan encontrar la solución más óptima a un problema mediante la codificación de poblaciones y cromosomas, en este caso el problema es un sudoku de 9x9 los cuales son grabados previamente en el sistema y que el usuario puede escoger entre diferentes dificultados, en mi caso implemente un sistema web con mediante HTML y CSS, para la codificación del programa utilice JavaScript, como editor de texto utilice Sublime Text, además de utilizar un servidor Web para la página Web siendo esta 000webhost.com, las pruebas las realice con el explorados Opera GX.

Link del sitio Web:

https://sudokuumg.000webhostapp.com/

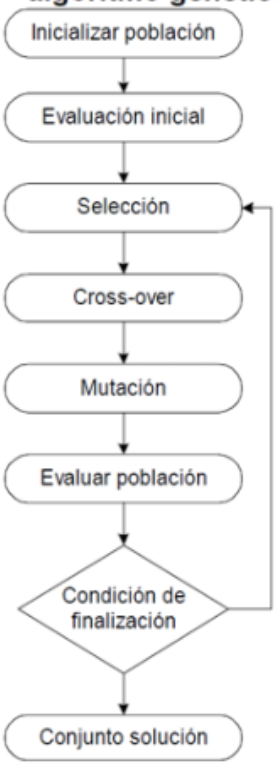
Proceso de Diseño e implementación.

Para el diseño del programa primero me dediqué a ver la información de la plataforma del curso y me guie principalmente del video "Algoritmos Genéticos en 5 minutos" en donde detallan el proceso para resolver un Sudoku de 4x4 comencé la codificación teniendo en cuanta los mismos procesos de selección, cruce de algoritmos, mutación y sustitución, utilicé la misma población estática de 10 cromosomas para resolver el problema y me atore un tiempo pensando cómo implementarlo correctamente en la programación puesto que no tenía claro cuáles son las selecciones y cruces en el caso del sudoku, considero que lo más complicado fue la abstracción de las idea a la codificación, hice algunos bosquejos y esquemas al iniciar para iniciar el proyecto los cuales los pegare en un anexo del presente informe.

OSCAR RODOLFO CHÁVEZ CAMEY 3090-19-13543 Noveno Ciclo

Utilice el siguiente esquema para guiarme en la programación y diseño del algoritmo:

Figura 2. Diagrama de funcionamiento de una posible implementación de algoritmo genético



Pasé un tiempo con el Sudoku de 4x4 y decidí cambiar el proceso al de 9x9 ya que no tenía claro como terminar el de 4x4 y el de 9x9 era la clave para el entregable del proyecto, en ese momento tenía problemas al abstraer el cromosoma padre para hacer el cruce puesto que al llenar el sudoku los números iniciales del algoritmo no debían ser alterados por el mismo hecho que son necesarios para la resolver correctamente el problema, esto lo solucione al momento de declarar el sudoku, para los espacios en blanco utilice ceros y una función guardara en otra variable los espacios que habían sido modificados, así al momento de extraerlos no tocaría los números iniciales del sudoku.

En la imagen muestro la forma del sudoku almacenado en el programa y como obtengo los espacios a evaluar con la función obtenerEv(), la función se encarga precisamente de lo siguiente:

```
switch (sudoku) {
  case 1:
    arrayA = [ 8, 4, 9,  0, 1, 2,  0, 0, 3,
               7, 2, 5,  4, 0, 0,  6, 0, 9,
               3, 6, 1,  9, 0, 5,  4, 0, 8,

               2, 9, 0,  0, 0, 0,  8, 0, 6,
               5, 0, 4,  0, 0, 0,  2, 0, 0,
               1, 0, 6,  0, 0, 0,  0, 3, 5,

               9, 0, 2,  8, 0, 3,  1, 6, 7,
               6, 0, 7,  0, 0, 9,  0, 0, 0,
               4, 0, 0,  1, 6, 0,  0, 9, 2];
    arrayEv = obtenerEv(arrayA);
    break;
  case 2:
    arrayA = [6, 5, 3,  1, 8, 7,  4, 0, 0,
              8, 9, 2,  6, 4, 3,  1, 7, 5,
              4, 7, 1,  9, 2, 5,  8, 3, 6,

              3, 0, 8,  0, 0, 9,  0, 0, 1,
              1, 0, 7,  0, 3, 2,  0, 0, 8,
              5, 0, 9,  8, 0, 1,  3, 4, 0,

              0, 3, 5,  0, 1, 4,  6, 8, 0,
              7, 1, 6,  3, 9, 8,  2, 5, 4,
              0, 8, 4,  0, 5, 6,  0, 1, 3];
    arrayEv = obtenerEv(arrayA);
    break;
```

Luego de rellenar el sudoku debía ser evaluado mediante la función fitness la cual mediría la aptitud de cada cromosoma para resolver el sudoku, me guie con el sudoku de 4x4 por lo que mi función Fitness evalúa cuantos errores existen por filas, columnas y cuadrantes, en el caso de los cuadrantes debe detectar que no haya varias colisiones dentro del sudoku, en ese caso contara dos errores en lugar de solo un error por cuadrante de 3x3. La función fitness es muy larga para añadirla en el documento, pero en resumen guarda las filas a hacer evaluadas, luego una función casi idéntica evalúa las columnas y por último la primera función evalúa los cuadrantes ya que recibe las posiciones de los array para cada caso. La razón por la que hay dos funciones parecidas es por la forma en la que presentaba los errores al inicio de la programación, ya que mostraba si encontraba errores en las filas y columnas mostrando X cuando contaba un error y O cuando no encontraba ninguno, luego fue actualizada a que mostrara más de un error por fila o columnas para mejorar la precisión del algoritmo, por último se eliminó que mostrara los errores en pantalla ya que solo mostraremos el número de cromosoma y el peso fitness como valor, ya no el sudoku entero debido a la gran impresión de caracteres por las generaciones.

Métodos de Selección

Ahora, para los medos de selección se plantearon los siguientes:

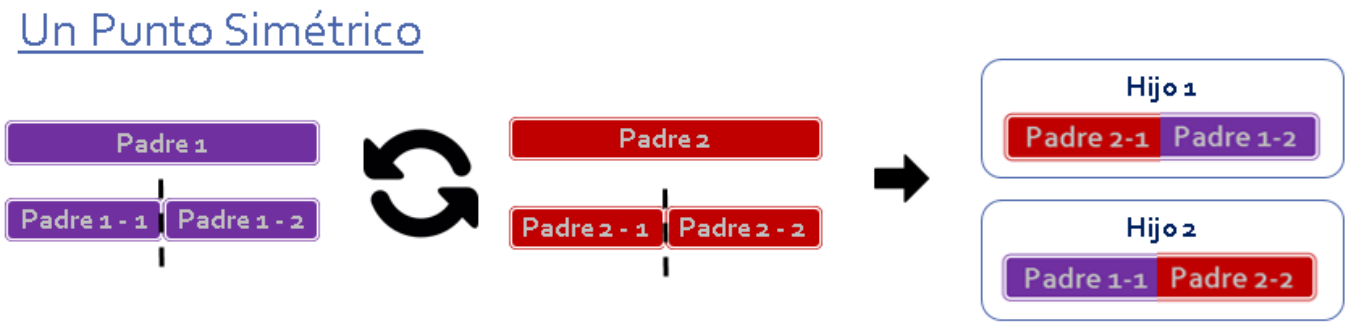
- Elitista:
  - Para este método de selección se ordenaron los cromosomas en orden ascendente de forma que el que tuviera un mayor peso fitness, es decir, el que tuviera más errores quedara hasta el final del podio, de los 10 cromosomas se seleccionan los 6 mejores tienen 4 hijos los cuales entran a la siguiente generación junto con los 6 mejores anteriores, es decir los padres de estos.

- Por Torneo:
  - En el caso del torneo escogen 20 cromosomas para que tengan 10 hijos de esta forma se reemplazan a la nueva generación (este método es el usado en el video del material de aprendizaje).
  - Para cada padre se seleccionan 3 cromosomas al azar y se elige al más apto.
- Generacional:
  - Se encarga que ningún cromosoma padre pase a la siguiente generación por lo que son reemplazados por los mejores cromosomas luego del método cruce, al ser 10 cromosomas se cruzan algunos padres nuevamente para que siempre existan 10 cromosomas en la siguiente generación.
- Especial:
  - Este caso es propio en el cual se combinan los 3 métodos de cruzamiento para los cromosomas y los ordena de forma elitista, es decir que solo los mejores pasan a la siguiente generación.

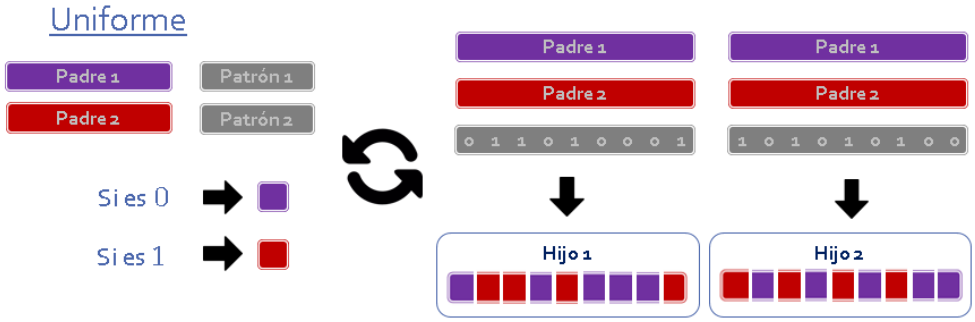
Métodos de Cruzamiento

Para los métodos de cruzamiento, hice lo siguiente:

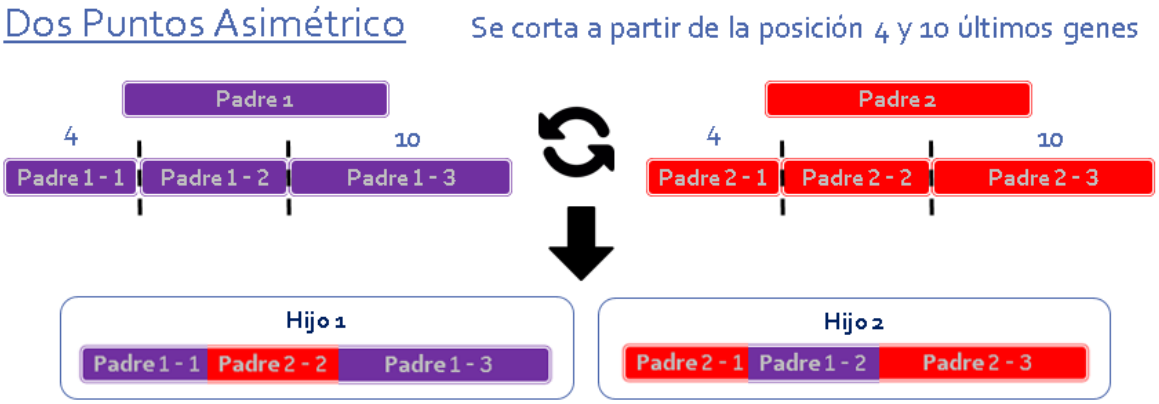
- Método de Cruzamiento 1 (Cruzamiento de 1 Puntos simétrico):
  - Método que toma como base la mitad del cromosoma de cada padre para crear a los hijos, se una la primera mitad de uno con la segunda mitad del otro y se escoge al más apto.



- Método de Cruzamiento 2 (Cruzamiento Uniforme):
  - Genera dos patrones de unos y ceros para crear a los hijos, tomando un gen de cada padre según sea 1 o 0, de esta forma se crean dos hijos, uno por cada patrón y se elige al más apto.



- Método de Cruzamiento 3(Cruzamiento de 2 puntos asimétrico):
  - En este método se utilizan las primeras 4 posiciones y las ultimas 10 de cada cromosoma, formando dos hijos, el primero con el primer tercio del primero el segundo tercio del segundo y el tercer tercio del primero, el segundo hijo es la inversa, el primer tercio del segundo el segundo tercio del primero y el tercer tercio del segundo. Se escoge el que resulte más apto.



OSCAR RODOLFO CHÁVEZ CAMEY 3090-19-13543 Noveno Ciclo

Mutación

La mutación se realizó de forma generacional, mutando los dos hijos de cada método de selección, al final se escoge al más apto.

Otras funciones dentro del código esta la función para mostrar el sudoku la cual lleva por nombre imprime DADO.

```
//Funciom para mostrar el sodoku inicial y final
function imprimeDADO(array) {

  let output = "<table>"; // Inicio de la tabla
  for (let i = 0; i < 81; i += 9) {
    output += "<tr><td>" + array.slice(i, i + 9).join("</td><td>") + "</td></tr>";
  }
  output += "</table>"; // Fin de la tabla
  document.write(output);
}
```

La función obtenerEv(array) se utiliza para conocer los espacios en los cual existen ceros, de esta forma se evita modificar el sudoku inicial y modificar únicamente los genes para la solución.

```
// Funcion para obtener los espacios de cada gen del cromosomas
function obtenerEv(array) {
  let nuevoArray = [];
  for (let i = 0; i < array.length; i++) { //Recorre todo el array
    if (array[i] === 0) {
      nuevoArray.push(i);   //Guarda la posicion de los 0s en un nuevo array
    }
  }
  return nuevoArray;        //Retorna el nuevo array con las posiciones de los genes
}
```

La función obtenerCromosoma(array) se utiliza para rellenar el sudoku reemplazando los ceros del sudoku seleccionado por números entre 1 y 9 previo a ser evaluado.

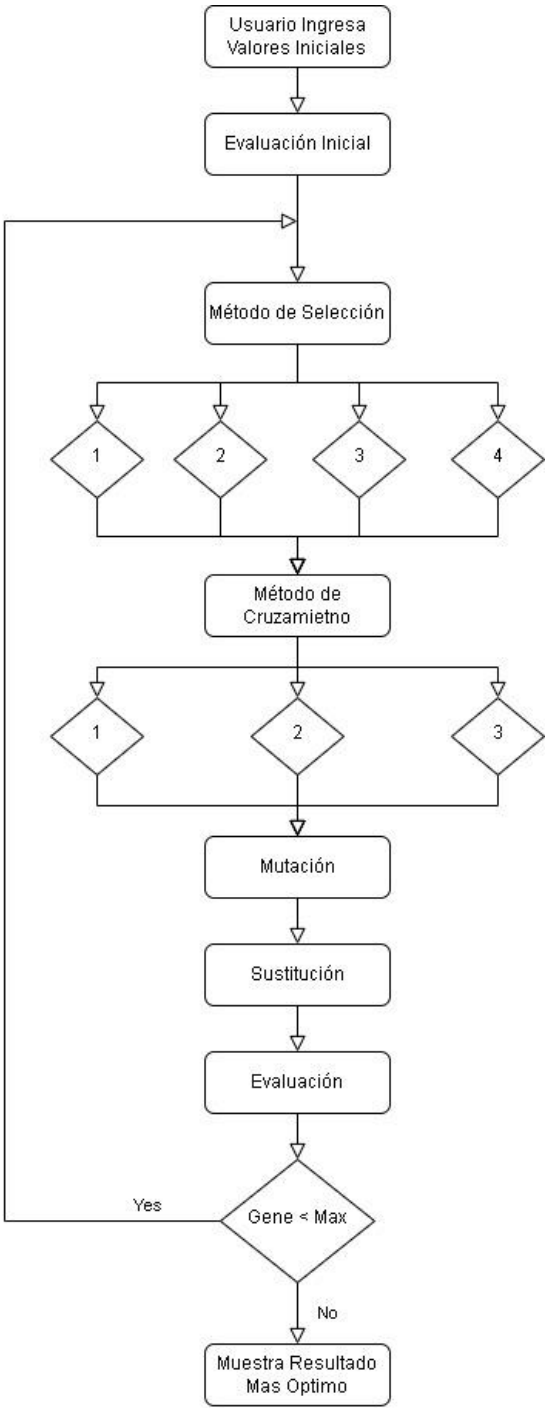
```
//Funcion para crear los cromosomas de la poblacion
function obtenerCromosoma(array) {
  let nuevoArray = [];
  for (let i = 0; i < array.length; i++) { //Recorre el sudoku en busca de los 0s
    if (array[i] === 0) {
      nuevoArray.push(Math.floor(Math.random() * 9) + 1); //Al encontrarlo lo reemplaza por numeros ente 1-9
    } else {
      nuevoArray.push(array[i]); //Los numeros del problema se copian nuevamenete
    }
  }
  return nuevoArray;

}
```

Criterio de Convergencia

Para la validación de solución del programa se recibe el parámetro generaciones iteradas con un máximo establecido por el usuario, en el sistema coloque como límite cien mil generaciones, aunque el explorados solo permite veinticinco mil generaciones sin mostrar el mensaje de poca memoria por lo que las pruebas de desempeño las realizare basándome en ese límite, el programa mostrara la solución más óptima al momento la cual será quien tenga mejor aptitud, es decir, la más cercana a 0 del podio de generaciones.

Además, el sistema se detendrá en el momento de encontrar la solución más óptima al problema la cual será la que tenga aptitud 0, esto interrumpirá las iteraciones y mostrara como resultado el sudoku resuelto.

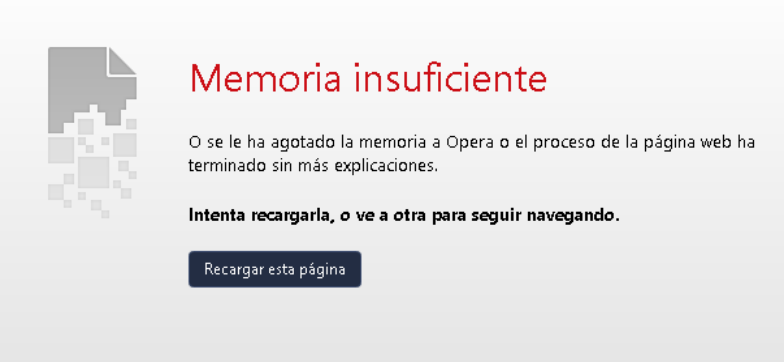
Diagrama General del Algoritmo



OSCAR RODOLFO CHÁVEZ CAMEY 3090-19-13543 Noveno Ciclo

Resultados de Pruebas

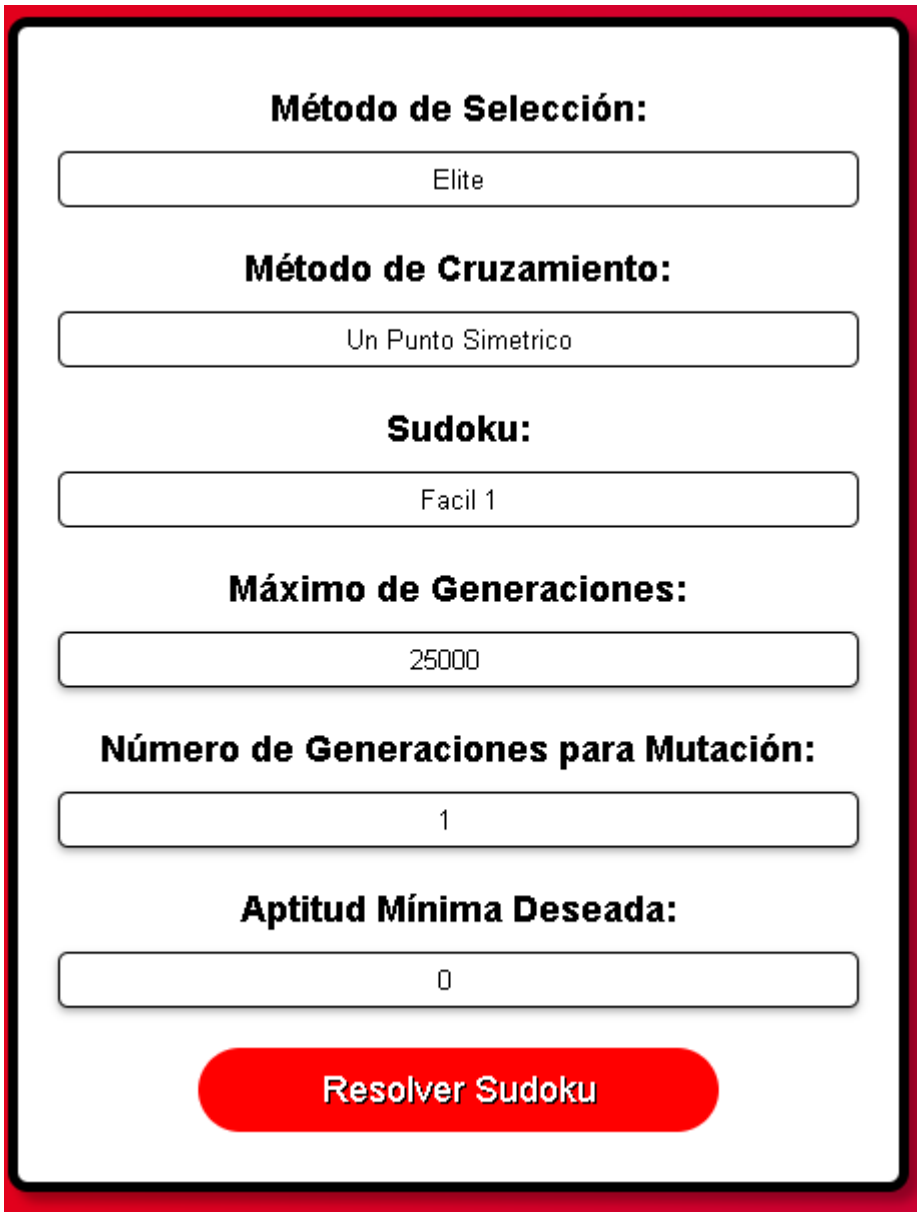
Durante el periodo de pruebas del programa utilicé el explorador Opera GX, el mayor inconveniente con el que me encontré es la falta de memoria para alimentar el proceso Web seguramente por la baja capacidad de mi equipo o por la cantidad de elementos abiertos al momento de su ejecución, debido a esto el mismo explorador suspendía el consumo de otras pestañas, pero el mensaje era el mismo:



Por lo que el informe de Pruebas se hará con 25 mil generaciones analizando los mínimos de aptitud

Pruebas Positivas

Sudoku 1:



Prueba 45:

GENERACION 652

Cromosoma 1
Fitness C1: 5

Cromosoma 2
Fitness C2: 0

Cromosoma 3
Fitness C3: 5

Cromosoma 4
Fitness C4: 7

Cromosoma 5
Fitness C5: 2

Cromosoma 6
Fitness C6: 3

Cromosoma 7
Fitness C7: 3

Cromosoma 8
Fitness C8: 3

Cromosoma 9
Fitness C9: 3

Cromosoma 10
Fitness C10: 3

RESULTADO

Cromosoma 2

8	4	9	6	1	2	7	5	3
7	2	5	4	3	8	6	1	9
3	6	1	9	7	5	4	2	8
2	9	3	7	5	1	8	4	6
5	8	4	3	9	6	2	7	1
1	7	6	2	8	4	9	3	5
9	5	2	8	4	3	1	6	7
6	1	7	5	2	9	3	8	4
4	3	8	1	6	7	5	9	2

Fitness C : 0

Prueba 49

GENERACION 930

Cromosoma 1
Fitness C1: 0

Cromosoma 2
Fitness C2: 5

Cromosoma 3
Fitness C3: 6

Cromosoma 4
Fitness C4: 5

Cromosoma 5
Fitness C5: 2

Cromosoma 6
Fitness C6: 3

Cromosoma 7
Fitness C7: 3

Cromosoma 8
Fitness C8: 3

Cromosoma 9
Fitness C9: 3

Cromosoma 10
Fitness C10: 3

RESULTADO

Cromosoma 1

8	4	9	6	1	2	7	5	3
7	2	5	4	3	8	6	1	9
3	6	1	9	7	5	4	2	8
2	9	3	7	5	1	8	4	6
5	8	4	3	9	6	2	7	1
1	7	6	2	8	4	9	3	5
9	5	2	8	4	3	1	6	7
6	1	7	5	2	9	3	8	4
4	3	8	1	6	7	5	9	2

Fitness C : 0

UNIVERSIDAD MARIANO GALVEZ DE GUATEMALA
CENTRO UNIVERSITARIO DE MAZATENANGO - SUCHITEPEQUEZ
FACULTAD DE INGENIERIA EN SISTEMAS
INTELIGENCIA ARTIFICIAL - Ing. Miguel Lemus

Prueba 161

GENERACION 10981

Cromosoma 1
Fitness C1: 5

Cromosoma 2
Fitness C2: 6

Cromosoma 3
Fitness C3: 6

Cromosoma 4
Fitness C4: 6

Cromosoma 5
Fitness C5: 6

Cromosoma 6
Fitness C6: 4

Cromosoma 7
Fitness C7: 6

Cromosoma 8
Fitness C8: 4

Cromosoma 9
Fitness C9: 5

Cromosoma 10
Fitness C10: 0

RESULTADO

Cromosoma 10

8	4	9	6	1	2	7	5	3
7	2	5	4	3	8	6	1	9
3	6	1	9	7	5	4	2	8
2	9	3	7	5	1	8	4	6
5	8	4	3	9	6	2	7	1
1	7	6	2	8	4	9	3	5
9	5	2	8	4	3	1	6	7
6	1	7	5	2	9	3	8	4
4	3	8	1	6	7	5	9	2

Fitness C : 0

OSCAR RODOLFO CHÁVEZ CAMEY 3090-19-13543 Noveno Ciclo

Prueba 244

GENERACION 11711

Cromosoma 1
Fitness C1: 0

Cromosoma 2
Fitness C2: 6

Cromosoma 3
Fitness C3: 5

Cromosoma 4
Fitness C4: 6

Cromosoma 5
Fitness C5: 6

Cromosoma 6
Fitness C6: 6

Cromosoma 7
Fitness C7: 6

Cromosoma 8
Fitness C8: 3

Cromosoma 9
Fitness C9: 3

Cromosoma 10
Fitness C10: 6

RESULTADO

Cromosoma 1

8	4	9	6	1	2	7	5	3
7	2	5	4	3	8	6	1	9
3	6	1	9	7	5	4	2	8
2	9	3	7	5	1	8	4	6
5	8	4	3	9	6	2	7	1
1	7	6	2	8	4	9	3	5
9	5	2	8	4	3	1	6	7
6	1	7	5	2	9	3	8	4
4	3	8	1	6	7	5	9	2

Fitness C : 0

OSCAR RODOLFO CHÁVEZ CAMEY 3090-19-13543 Noveno Ciclo

UNIVERSIDAD MARIANO GALVEZ DE GUATEMALA

CENTRO UNIVERSITARIO DE MAZATENANGO - SUCHITEPEQUEZ

FACULTAD DE INGENIERIA EN SISTEMAS

INTELIGENCIA ARTIFICIAL - Ing. Miguel Lemus

Numero 244

GENERACION 7792

Cromosoma 1
Fitness C1: 0

Cromosoma 2
Fitness C2: 2

Cromosoma 3
Fitness C3: 2

Cromosoma 4
Fitness C4: 5

Cromosoma 5
Fitness C5: 2

Cromosoma 6
Fitness C6: 2

Cromosoma 7
Fitness C7: 2

Cromosoma 8
Fitness C8: 2

Cromosoma 9
Fitness C9: 4

Cromosoma 10
Fitness C10: 2

RESULTADO

Cromosoma 1

8	4	9	6	1	2	7	5	3
7	2	5	4	3	8	6	1	9
3	6	1	9	7	5	4	2	8
2	9	3	7	5	1	8	4	6
5	8	4	3	9	6	2	7	1
1	7	6	2	8	4	9	3	5
9	5	2	8	4	3	1	6	7
6	1	7	5	2	9	3	8	4
4	3	8	1	6	7	5	9	2

Fitness C : 0

OSCAR RODOLFO CHÁVEZ CAMEY 3090-19-13543 Noveno Ciclo

Conclusiones Obtenidas

Los algoritmos genéticos permiten encontrar la solución al problema de solución del sudoku 9x9 métodos más variados como lo es el cruce uniforme y la selección por torneo obtienen mejores resultados.

La mutación juega un papel importante cuando a través de las generaciones se estanca en un mismo cromosoma por lo que pare este caso es recomendable elevar la cantidad de veces que se realiza la mutación, el detalle es que esto afecta demasiado al cromosoma.

Este tipo de programas que reciben muchas iteraciones en el caso de la implementación, opino que es mejor hacerlo en un compilador por el uso de memoria del explorador o bien investigar formas de optimización de código.

OSCAR RODOLFO CHÁVEZ CAMEY 3090-19-13543 Noveno Ciclo