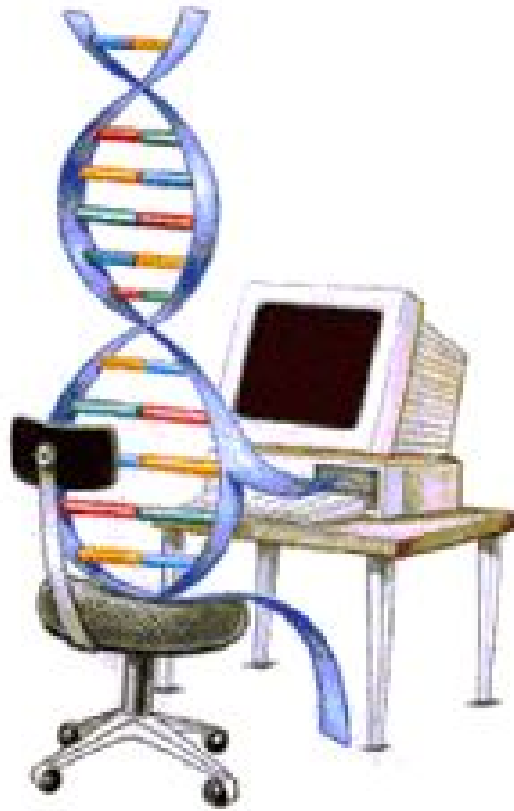


Práctica 1: algoritmo genético

Sistemas Inteligentes - 3º Ingeniería Informática - 2016/2017



Jorge Gallego Madrid

Grupo 1.2
[12-11-2016]

Algoritmo genético	2
Cuestiones para el diseño e implementación	3
1) Añadir la definición del genoma en “Sudoku.cpp”.	3
2) Añadir el casting en el sitio indicado en la función InicioSudoku().	3
3) Dada esa definición, explica de qué manera se están inicializando los individuos en el AG	3
propuesto.	3
4) Explica el funcionamiento de los operadores de selección indicados en la sección “Ajuste del	3
Algoritmo Genético”.	3
5) Insértalos de la manera adecuada en “Sudoku.cpp”.	3
6) Escribir el casting en el sitio indicado en la función CruceSudoku().	4
7) Dada esa definición, explica de qué manera se están cruzando los individuos en el AG	4
propuesto.	4
8) Escribir el casting en el sitio indicado en la función MutacionSudoku().	4
9) Dada esa definición, explica de qué manera se están mutando los individuos en el AG	4
propuesto.	4
10) Define y explica la condición de parada que utilizarás.	5
11) Implementar la condición de parada en “Sudoku.cpp”.	5
12) Diseña y explica la función fitness que utilizarás. Recuerda, como se indica al comienzo de este	5
guión, que una solución del sudoku no puede repetir en una misma fila, columna o	5
subcuadrícula ninguno de los números	5
13) Implementar la función fitness en “Sudoku.cpp”.	6
14) Completar con la implementación de los elementos necesarios para cubrir los objetivos del AG propuesto.	7
Tabla de valores de la función fitness	9
Análisis de las pruebas de ajuste	19
Manual y asignación	20
Protocolo de resolución de los casos de usuario	21
Bibliografía	24

Algoritmo genético

Un algoritmo genético es una variante de la búsqueda de haz estocástica en el que los estados sucesores se generan combinando a dos estados padres, en mayor medida que modificando un solo estado.

Los algoritmos genéticos comienzan con un conjunto de estados generados aleatoriamente, a esto lo llamamos población. Cada estado o individuo, está codificado como una cadena sobre un alfabeto finito. A su vez, cada elemento de un individuo, es denominado gen. Se necesita un conjunto de operadores genéticos para la creación de nuevos individuos.

El valor heurístico de un estado es denominado fitness o fenotipo. La función que calcula este valor mide la calidad de los individuos, con el objetivo de seleccionarlos y combinarlos posteriormente para así obtener estados mejores.

El procedimiento que sigue un algoritmo genético empieza con una población de k estados generados aleatoriamente. Seguidamente comienza el bucle principal que itera hasta que se alcanza la condición de parada o hasta que se alcanza el número máximo de generaciones. Dentro de este bucle, se seleccionan k individuos de la población actual para crear la población intermedia y, de éstos, se eligen los individuos para ser cruzados. Los individuos elegidos, que han sido seleccionados con una probabilidad p_c , se emparejan y se les aplica un operador de cruce. Estos nuevos individuos sustituirán a los padres. Antes de terminar la iteración, con una probabilidad p_m , se mutan los genes de los individuos de la población actual. Finalmente, esta nueva población formará la población inicial de la siguiente generación.

Cuestiones para el diseño e implementación

1) Añadir la definición del genoma en “Sudoku.cpp”.

Añadimos:

```
GA1DArrayAlleleGenome<int> & genome.
```

2) Añadir el casting en el sitio indicado en la función InicioSudoku().

Hacemos el casting:

```
GA1DArrayAlleleGenome<int> &genome = (GA1DArrayAlleleGenome<int> &) g;
```

3) Dada esa definición, explica de qué manera se están inicializando los individuos en el AG propuesto.

El algoritmo recorre mediante la función InicioSudoku() todas las filas de la plantilla, que representan en su conjunto a un individuo. Para cada una de ellas, con ayuda de una estructura auxiliar, se buscan los fijos y se colocan en dicha estructura, que previamente ha sido inicializada sin repetidos. Cuando se encuentra un fijo en la plantilla, éste se busca en la estructura auxiliar y se intercambia a su posición correcta, evitando de esta manera que se produzcan repeticiones innecesarias. Finalmente, se copia la fila en el genoma.

4) Explica el funcionamiento de los operadores de selección indicados en la sección “Ajuste del Algoritmo Genético”.

Aparecen indicados dos operadores de selección:

- Ruleta: se eligen los individuos con probabilidad proporcional a su función de idoneidad.
- Torneo: se establecen k torneos aleatorios entre parejas de individuos y se eligen los que ganan en cada torneo (mejor función de idoneidad).

5) Insértalos de la manera adecuada en “Sudoku.cpp”.

Introducimos un if-else, donde en función del parámetro pasado en tercer lugar, se selecciona uno (“ruleta”) u otro (“torneo”).

6) Escribir el casting en el sitio indicado en la función `CruceSudoku()`.

Se escriben 4 casting en total, en los lugares indicados. Son los siguientes:

```
// Madre
GA1DArrayAlleleGenome<int> &m = (GA1DArrayAlleleGenome<int> &) p1;
// Padre
GA1DArrayAlleleGenome<int> &p = (GA1DArrayAlleleGenome<int> &) p2;

// Hijo1
GA1DArrayGenome<int> &h1 = (GA1DArrayGenome<int> &)*c1;
// Hijo2
GA1DArrayGenome<int> &h2 = (GA1DArrayGenome<int> &)*c2;
```

7) Dada esa definición, explica de qué manera se están cruzando los individuos en el AG propuesto.

Se obtiene la plantilla de la madre y se busca un punto de cruce que tiene que corresponder con el inicio de una fila en el individuo, este será el punto uno. El segundo punto de cruce se obtendrá restando el punto uno a la longitud total del array que representa al individuo. Una vez hecho esto, el hijo1 hereda la primera parte de la madre y la segunda parte del padre y el hijo2 hereda la primera parte del padre y la segunda parte de la madre.

8) Escribir el casting en el sitio indicado en la función `MutacionSudoku()`.

Añadimos el siguiente casting:

```
GA1DArrayAlleleGenome<int> &genome = (GA1DArrayAlleleGenome<int> &) g
```

9) Dada esa definición, explica de qué manera se están mutando los individuos en el AG propuesto.

Recorremos el sudoku y en función de la probabilidad de mutación, ésta va a tener lugar o no. Cuando toca mutar, elegimos fila o columna de forma aleatoria 50/50 y comenzamos la mutación. Comprobamos si hay repetidos y generamos un nuevo valor que no existe, que colocaremos en una posición donde exista un repetido. Una vez hecho esto, ponemos el repetido en el lugar donde estaba el generado, arreglando la fila. En caso de que fuera un fijo, deshacemos la mutación.

10) Define y explica la condición de parada que utilizarás.

Tendremos dos condiciones de parada:

- Si el algoritmo encuentra un valor de fitness igual a 0 podemos parar, ya que eso significa que ya tenemos una solución y no es necesario seguir iterando.
- Si hemos alcanzado el número máximo de generaciones hemos de parar y dar como resultado el mejor valor de fitness que hayamos encontrado.

11) Implementar la condición de parada en “Sudoku.cpp”.

```
// Funcion de terminacion
GABoolean Termina(GAGeneticAlgorithm & ga){
    // Para si encuentra el 0 (sudoku solucion) o si
    // llegamos al limite de generaciones (12000)
    if ((ga.statistics().minEver()==0) ||
        (ga.statistics().generation()==ga.nGenerations()))
        return gaTrue;
    else
        return gaFalse;
}
```

12) Diseña y explica la función fitness que utilizarás. Recuerda, como se indica al comienzo de este guión, que una solución del sudoku no puede repetir en una misma fila, columna o subcuadrícula ninguno de los números

Usaremos una función que se divide en tres partes y que se ayuda de una adaptación de la función checkColumn, proporcionada por el profesor. Esta adaptación, en vez de devolver un booleano si hay repeticiones, devuelve el número de repeticiones en el array que se le ha pasado como parámetro.

Cada una de las partes de la función fitness (Objective) se encarga de contar el número de repeticiones primero en filas, luego en columnas y, finalmente, en cuadrículas. Devuelve el número total de repeticiones encontradas en el sudoku.

13) Implementar la función fitness en "Sudoku.cpp".

```
// Funcion auxiliar para calcular las repeticiones
// dentro de un array. Modificada a partir de checkColumna
float cuentaRepeticiones(int col[], int * check, int tam){
    int repe = 0;
    for(int i=0;i<tam;i++)
        check[i]=0;
    for(int i=0;i<tam;i++)
        check[col[i]-1]++;
    for(int i=0;i<tam;i++)
        if (check[i]>1)
            repe += check[i]-1;
    return repe;
}

// Funcion objetivo
// Numero de repeticiones que se encuentran
float Objective(GAGenome& g) {
    GA1DArrayAlleleGenome<int> & genome = (GA1DArrayAlleleGenome<int>
&)g;

    struct plantilla * plant;
    plant = (struct plantilla *) genome.userData();
    float repeticiones = 0;

    // Comprobamos repeticiones en filas
    int aux[plant->tam];
    int * check = new int[plant->tam];
    // Para cada fila
    for (int i = 0; i < plant->tam*plant->tam; i+=plant->tam){
        // Relleno aux con sus valores
        for (int j = 0; j < plant->tam; j++){
            aux[j] = genome.gene(j+i);
        }
        // Se la paso a cuentaRepeticiones
        repeticiones += cuentaRepeticiones(aux, check, plant->tam);
    }

    // Comprobamos repeticiones en columnas
    // Para cada columna
    for (int i = 0; i < plant->tam; i++){
        // Relleno aux con sus valores
        for (int j = i; j < plant->tam*plant->tam; j+=plant->tam){
            aux[j/plant->tam] = genome.gene(j);
        }
    }
}
```

```

        // Se lo paso a cuentaRepeticiones
        repeticiones += cuentaRepeticiones(aux, check, plant->tam);
    }

    // Comprobamos repeticiones en cuadrículas
    int ind = 0;
    // Para cada esquina superior izquierda de una cuadrícula
    for (int i = 0; i < plant->tam*plant->tam; i+=3){
        // Lo recorro por filas
        for (int j = i; j < (i+plant->tam*3); j+=plant->tam){
            // Me situo en la columna
            for(int k = j; k < j+3; k++){
                aux[ind] = genome.gene(k);
                ind++;
            }
        }
        // Si llego a la cuadrícula mas a la derecha
        // salto a la siguiente de la izquierda más abajo
        if(i%plant->tam == plant->tam-3){
            i+=plant->tam*2;
        }
        // Se la paso a cuentaRepeticiones
        repeticiones += cuentaRepeticiones(aux, check, plant->tam);
        ind = 0;
    }
    return repeticiones;
}

```

14) Completar con la implementación de los elementos necesarios para cubrir los objetivos del AG propuesto.

En este punto, lo único que queda para completar el programa es implementar el main.

- Declaramos las variables para los parámetros del GA y las inicializamos con los parámetros. Además, hacemos una comprobación de posibles errores.
- Mostramos por pantalla la información del algoritmo.
- Creamos la plantilla y leemos el sudoku que nos indican.
- Definimos el conjunto de alelos, que serán enteros desde 1 a 9.
- Creamos el genoma, indicando el tamaño del vector, los alelos, la función objetivo y la plantilla donde están los datos. También le tenemos que decir las funciones que usaremos para la inicialización, el cruce y la mutación.
- En este punto ya podemos definir el tipo de algoritmo y que el objetivo va a ser minimizar el valor de fitness.

- Comunicamos al algoritmo los distintos parámetros, así como la función de terminación. Para que el AG tenga un comportamiento estable con respecto al generador de números aleatorios, invocaremos en el código al método `evolve` con valor 1 (`evolve(1)`).
- Finalmente, imprimimos el sudoku solución junto con el valor de fitness que hemos encontrado.

Tabla de valores de la función fitness

Las filas marcadas en gris son aquellas en las que esa configuración nos da un resultado de fitness igual a 0 para todos los sudokus de prueba.

Sudoku	Tam. pobl	Selector	pcruce	pmut	fitness
A1.txt	100	ruleta	0.8	0.01	6
A1.txt	100	ruleta	0.8	0.05	2
A1.txt	100	ruleta	0.8	0.1	4
A1.txt	100	ruleta	0.8	0.125	0
A1.txt	100	ruleta	0.8	0.15	0
A1.txt	100	ruleta	0.85	0.01	0
A1.txt	100	ruleta	0.85	0.05	5
A1.txt	100	ruleta	0.85	0.1	0
A1.txt	100	ruleta	0.85	0.125	4
A1.txt	100	ruleta	0.85	0.15	0
A1.txt	100	ruleta	0.9	0.01	4
A1.txt	100	ruleta	0.9	0.05	4
A1.txt	100	ruleta	0.9	0.1	4
A1.txt	100	ruleta	0.9	0.125	5
A1.txt	100	ruleta	0.9	0.15	0
A1.txt	100	ruleta	0.95	0.01	0
A1.txt	100	ruleta	0.95	0.05	4
A1.txt	100	ruleta	0.95	0.1	0
A1.txt	100	ruleta	0.95	0.125	0
A1.txt	100	ruleta	0.95	0.15	0
A1.txt	150	ruleta	0.8	0.01	0
A1.txt	150	ruleta	0.8	0.05	6
A1.txt	150	ruleta	0.8	0.1	0
A1.txt	150	ruleta	0.8	0.125	4
A1.txt	150	ruleta	0.8	0.15	2
A1.txt	150	ruleta	0.85	0.01	0
A1.txt	150	ruleta	0.85	0.05	0
A1.txt	150	ruleta	0.85	0.1	6
A1.txt	150	ruleta	0.85	0.125	0
A1.txt	150	ruleta	0.85	0.15	4
A1.txt	150	ruleta	0.9	0.01	0
A1.txt	150	ruleta	0.9	0.05	0
A1.txt	150	ruleta	0.9	0.1	0
A1.txt	150	ruleta	0.9	0.125	0
A1.txt	150	ruleta	0.9	0.15	0
A1.txt	150	ruleta	0.95	0.01	0
A1.txt	150	ruleta	0.95	0.05	0
A1.txt	150	ruleta	0.95	0.1	0
A1.txt	150	ruleta	0.95	0.125	4
A1.txt	150	ruleta	0.95	0.15	0

Sudoku	Tam. pobl	Selector	pcruce	pmut	fitness
A1.txt	100	torneo	0.8	0.01	10
A1.txt	100	torneo	0.8	0.05	0
A1.txt	100	torneo	0.8	0.1	4
A1.txt	100	torneo	0.8	0.125	0
A1.txt	100	torneo	0.8	0.15	6
A1.txt	100	torneo	0.85	0.01	6
A1.txt	100	torneo	0.85	0.05	0
A1.txt	100	torneo	0.85	0.1	0
A1.txt	100	torneo	0.85	0.125	0
A1.txt	100	torneo	0.85	0.15	4
A1.txt	100	torneo	0.9	0.01	0
A1.txt	100	torneo	0.9	0.05	0
A1.txt	100	torneo	0.9	0.1	0
A1.txt	100	torneo	0.9	0.125	0
A1.txt	100	torneo	0.9	0.15	0
A1.txt	100	torneo	0.95	0.01	8
A1.txt	100	torneo	0.95	0.05	0
A1.txt	100	torneo	0.95	0.1	0
A1.txt	100	torneo	0.95	0.125	0
A1.txt	100	torneo	0.95	0.15	0
A1.txt	150	torneo	0.8	0.01	10
A1.txt	150	torneo	0.8	0.05	4
A1.txt	150	torneo	0.8	0.1	0
A1.txt	150	torneo	0.8	0.125	0
A1.txt	150	torneo	0.8	0.15	0
A1.txt	150	torneo	0.85	0.01	6
A1.txt	150	torneo	0.85	0.05	0
A1.txt	150	torneo	0.85	0.1	0
A1.txt	150	torneo	0.85	0.125	0
A1.txt	150	torneo	0.85	0.15	0
A1.txt	150	torneo	0.9	0.01	2
A1.txt	150	torneo	0.9	0.05	4
A1.txt	150	torneo	0.9	0.1	0
A1.txt	150	torneo	0.9	0.125	0
A1.txt	150	torneo	0.9	0.15	5
A1.txt	150	torneo	0.95	0.01	2
A1.txt	150	torneo	0.95	0.05	0
A1.txt	150	torneo	0.95	0.1	0
A1.txt	150	torneo	0.95	0.125	0
A1.txt	150	torneo	0.95	0.15	0

Sudoku	Tam. pobl	Selector	pcruce	pmut	fitness
A2.txt	100	ruleta	0.8	0.01	0
A2.txt	100	ruleta	0.8	0.05	0
A2.txt	100	ruleta	0.8	0.1	3
A2.txt	100	ruleta	0.8	0.125	2
A2.txt	100	ruleta	0.8	0.15	2
A2.txt	100	ruleta	0.85	0.01	4
A2.txt	100	ruleta	0.85	0.05	2
A2.txt	100	ruleta	0.85	0.1	4
A2.txt	100	ruleta	0.85	0.125	0
A2.txt	100	ruleta	0.85	0.15	2
A2.txt	100	ruleta	0.9	0.01	0
A2.txt	100	ruleta	0.9	0.05	3
A2.txt	100	ruleta	0.9	0.1	4
A2.txt	100	ruleta	0.9	0.125	3
A2.txt	100	ruleta	0.9	0.15	0
A2.txt	100	ruleta	0.95	0.01	3
A2.txt	100	ruleta	0.95	0.05	3
A2.txt	100	ruleta	0.95	0.1	0
A2.txt	100	ruleta	0.95	0.125	2
A2.txt	100	ruleta	0.95	0.15	2
A2.txt	150	ruleta	0.8	0.01	0
A2.txt	150	ruleta	0.8	0.05	0
A2.txt	150	ruleta	0.8	0.1	2
A2.txt	150	ruleta	0.8	0.125	0
A2.txt	150	ruleta	0.8	0.15	0
A2.txt	150	ruleta	0.85	0.01	2
A2.txt	150	ruleta	0.85	0.05	4
A2.txt	150	ruleta	0.85	0.1	2
A2.txt	150	ruleta	0.85	0.125	5
A2.txt	150	ruleta	0.85	0.15	2
A2.txt	150	ruleta	0.9	0.01	7
A2.txt	150	ruleta	0.9	0.05	0
A2.txt	150	ruleta	0.9	0.1	2
A2.txt	150	ruleta	0.9	0.125	2
A2.txt	150	ruleta	0.9	0.15	3
A2.txt	150	ruleta	0.95	0.01	2
A2.txt	150	ruleta	0.95	0.05	2
A2.txt	150	ruleta	0.95	0.1	2
A2.txt	150	ruleta	0.95	0.125	4
A2.txt	150	ruleta	0.95	0.15	0

Sudoku	Tam. pobl	Selector	pcruce	pmut	fitness
A2.txt	100	torneo	0.8	0.01	4
A2.txt	100	torneo	0.8	0.05	0
A2.txt	100	torneo	0.8	0.1	0
A2.txt	100	torneo	0.8	0.125	0
A2.txt	100	torneo	0.8	0.15	0
A2.txt	100	torneo	0.85	0.01	8
A2.txt	100	torneo	0.85	0.05	0
A2.txt	100	torneo	0.85	0.1	0
A2.txt	100	torneo	0.85	0.125	0
A2.txt	100	torneo	0.85	0.15	0
A2.txt	100	torneo	0.9	0.01	4
A2.txt	100	torneo	0.9	0.05	0
A2.txt	100	torneo	0.9	0.1	2
A2.txt	100	torneo	0.9	0.125	0
A2.txt	100	torneo	0.9	0.15	2
A2.txt	100	torneo	0.95	0.01	5
A2.txt	100	torneo	0.95	0.05	0
A2.txt	100	torneo	0.95	0.1	2
A2.txt	100	torneo	0.95	0.125	0
A2.txt	100	torneo	0.95	0.15	0
A2.txt	150	torneo	0.8	0.01	2
A2.txt	150	torneo	0.8	0.05	0
A2.txt	150	torneo	0.8	0.1	0
A2.txt	150	torneo	0.8	0.125	0
A2.txt	150	torneo	0.8	0.15	0
A2.txt	150	torneo	0.85	0.01	4
A2.txt	150	torneo	0.85	0.05	0
A2.txt	150	torneo	0.85	0.1	0
A2.txt	150	torneo	0.85	0.125	0
A2.txt	150	torneo	0.85	0.15	0
A2.txt	150	torneo	0.9	0.01	8
A2.txt	150	torneo	0.9	0.05	2
A2.txt	150	torneo	0.9	0.1	0
A2.txt	150	torneo	0.9	0.125	0
A2.txt	150	torneo	0.9	0.15	3
A2.txt	150	torneo	0.95	0.01	4
A2.txt	150	torneo	0.95	0.05	2
A2.txt	150	torneo	0.95	0.1	0
A2.txt	150	torneo	0.95	0.125	0
A2.txt	150	torneo	0.95	0.15	0

Sudoku	Tam. pobl	Selector	pcruce	pmut	fitness
A3.txt	100	ruleta	0.8	0.01	0
A3.txt	100	ruleta	0.8	0.05	6
A3.txt	100	ruleta	0.8	0.1	6
A3.txt	100	ruleta	0.8	0.125	2
A3.txt	100	ruleta	0.8	0.15	4
A3.txt	100	ruleta	0.85	0.01	0
A3.txt	100	ruleta	0.85	0.05	0
A3.txt	100	ruleta	0.85	0.1	0
A3.txt	100	ruleta	0.85	0.125	4
A3.txt	100	ruleta	0.85	0.15	4
A3.txt	100	ruleta	0.9	0.01	0
A3.txt	100	ruleta	0.9	0.05	0
A3.txt	100	ruleta	0.9	0.1	4
A3.txt	100	ruleta	0.9	0.125	0
A3.txt	100	ruleta	0.9	0.15	8
A3.txt	100	ruleta	0.95	0.01	0
A3.txt	100	ruleta	0.95	0.05	2
A3.txt	100	ruleta	0.95	0.1	6
A3.txt	100	ruleta	0.95	0.125	0
A3.txt	100	ruleta	0.95	0.15	0
A3.txt	150	ruleta	0.8	0.01	0
A3.txt	150	ruleta	0.8	0.05	0
A3.txt	150	ruleta	0.8	0.1	0
A3.txt	150	ruleta	0.8	0.125	4
A3.txt	150	ruleta	0.8	0.15	0
A3.txt	150	ruleta	0.85	0.01	0
A3.txt	150	ruleta	0.85	0.05	0
A3.txt	150	ruleta	0.85	0.1	2
A3.txt	150	ruleta	0.85	0.125	4
A3.txt	150	ruleta	0.85	0.15	4
A3.txt	150	ruleta	0.9	0.01	4
A3.txt	150	ruleta	0.9	0.05	0
A3.txt	150	ruleta	0.9	0.1	4
A3.txt	150	ruleta	0.9	0.125	0
A3.txt	150	ruleta	0.9	0.15	6
A3.txt	150	ruleta	0.95	0.01	0
A3.txt	150	ruleta	0.95	0.05	2
A3.txt	150	ruleta	0.95	0.1	0
A3.txt	150	ruleta	0.95	0.125	4
A3.txt	150	ruleta	0.95	0.15	0

Sudoku	Tam. pobl	Selector	pcruce	pmut	fitness
A3.txt	100	torneo	0.8	0.01	4
A3.txt	100	torneo	0.8	0.05	0
A3.txt	100	torneo	0.8	0.1	0
A3.txt	100	torneo	0.8	0.125	0
A3.txt	100	torneo	0.8	0.15	0
A3.txt	100	torneo	0.85	0.01	6
A3.txt	100	torneo	0.85	0.05	0
A3.txt	100	torneo	0.85	0.1	0
A3.txt	100	torneo	0.85	0.125	0
A3.txt	100	torneo	0.85	0.15	0
A3.txt	100	torneo	0.9	0.01	2
A3.txt	100	torneo	0.9	0.05	0
A3.txt	100	torneo	0.9	0.1	0
A3.txt	100	torneo	0.9	0.125	0
A3.txt	100	torneo	0.9	0.15	0
A3.txt	100	torneo	0.95	0.01	8
A3.txt	100	torneo	0.95	0.05	0
A3.txt	100	torneo	0.95	0.1	0
A3.txt	100	torneo	0.95	0.125	0
A3.txt	100	torneo	0.95	0.15	0
A3.txt	150	torneo	0.8	0.01	9
A3.txt	150	torneo	0.8	0.05	0
A3.txt	150	torneo	0.8	0.1	0
A3.txt	150	torneo	0.8	0.125	0
A3.txt	150	torneo	0.8	0.15	0
A3.txt	150	torneo	0.85	0.01	3
A3.txt	150	torneo	0.85	0.05	0
A3.txt	150	torneo	0.85	0.1	0
A3.txt	150	torneo	0.85	0.125	0
A3.txt	150	torneo	0.85	0.15	0
A3.txt	150	torneo	0.9	0.01	2
A3.txt	150	torneo	0.9	0.05	0
A3.txt	150	torneo	0.9	0.1	0
A3.txt	150	torneo	0.9	0.125	0
A3.txt	150	torneo	0.9	0.15	0
A3.txt	150	torneo	0.95	0.01	4
A3.txt	150	torneo	0.95	0.05	0
A3.txt	150	torneo	0.95	0.1	0
A3.txt	150	torneo	0.95	0.125	0
A3.txt	150	torneo	0.95	0.15	0

Sudoku	Tam. pobl	Selector	pcruce	pmut	fitness
A4.txt	100	ruleta	0.8	0.01	0
A4.txt	100	ruleta	0.8	0.05	2
A4.txt	100	ruleta	0.8	0.1	0
A4.txt	100	ruleta	0.8	0.125	2
A4.txt	100	ruleta	0.8	0.15	2
A4.txt	100	ruleta	0.85	0.01	0
A4.txt	100	ruleta	0.85	0.05	0
A4.txt	100	ruleta	0.85	0.1	0
A4.txt	100	ruleta	0.85	0.125	0
A4.txt	100	ruleta	0.85	0.15	2
A4.txt	100	ruleta	0.9	0.01	0
A4.txt	100	ruleta	0.9	0.05	4
A4.txt	100	ruleta	0.9	0.1	0
A4.txt	100	ruleta	0.9	0.125	2
A4.txt	100	ruleta	0.9	0.15	2
A4.txt	100	ruleta	0.95	0.01	2
A4.txt	100	ruleta	0.95	0.05	2
A4.txt	100	ruleta	0.95	0.1	2
A4.txt	100	ruleta	0.95	0.125	2
A4.txt	100	ruleta	0.95	0.15	2
A4.txt	150	ruleta	0.8	0.01	2
A4.txt	150	ruleta	0.8	0.05	2
A4.txt	150	ruleta	0.8	0.1	0
A4.txt	150	ruleta	0.8	0.125	2
A4.txt	150	ruleta	0.8	0.15	2
A4.txt	150	ruleta	0.85	0.01	0
A4.txt	150	ruleta	0.85	0.05	0
A4.txt	150	ruleta	0.85	0.1	0
A4.txt	150	ruleta	0.85	0.125	2
A4.txt	150	ruleta	0.85	0.15	4
A4.txt	150	ruleta	0.9	0.01	0
A4.txt	150	ruleta	0.9	0.05	0
A4.txt	150	ruleta	0.9	0.1	0
A4.txt	150	ruleta	0.9	0.125	2
A4.txt	150	ruleta	0.9	0.15	4
A4.txt	150	ruleta	0.95	0.01	2
A4.txt	150	ruleta	0.95	0.05	2
A4.txt	150	ruleta	0.95	0.1	2
A4.txt	150	ruleta	0.95	0.125	2
A4.txt	150	ruleta	0.95	0.15	2

Sudoku	Tam. pobl	Selector	pcruce	pmut	fitness
A4.txt	100	torneo	0.8	0.01	3
A4.txt	100	torneo	0.8	0.05	0
A4.txt	100	torneo	0.8	0.1	0
A4.txt	100	torneo	0.8	0.125	0
A4.txt	100	torneo	0.8	0.15	0
A4.txt	100	torneo	0.85	0.01	2
A4.txt	100	torneo	0.85	0.05	0
A4.txt	100	torneo	0.85	0.1	0
A4.txt	100	torneo	0.85	0.125	2
A4.txt	100	torneo	0.85	0.15	0
A4.txt	100	torneo	0.9	0.01	4
A4.txt	100	torneo	0.9	0.05	0
A4.txt	100	torneo	0.9	0.1	0
A4.txt	100	torneo	0.9	0.125	0
A4.txt	100	torneo	0.9	0.15	0
A4.txt	100	torneo	0.95	0.01	8
A4.txt	100	torneo	0.95	0.05	2
A4.txt	100	torneo	0.95	0.1	0
A4.txt	100	torneo	0.95	0.125	0
A4.txt	100	torneo	0.95	0.15	0
A4.txt	150	torneo	0.8	0.01	4
A4.txt	150	torneo	0.8	0.05	0
A4.txt	150	torneo	0.8	0.1	0
A4.txt	150	torneo	0.8	0.125	0
A4.txt	150	torneo	0.8	0.15	0
A4.txt	150	torneo	0.85	0.01	10
A4.txt	150	torneo	0.85	0.05	0
A4.txt	150	torneo	0.85	0.1	0
A4.txt	150	torneo	0.85	0.125	0
A4.txt	150	torneo	0.85	0.15	0
A4.txt	150	torneo	0.9	0.01	6
A4.txt	150	torneo	0.9	0.05	0
A4.txt	150	torneo	0.9	0.1	0
A4.txt	150	torneo	0.9	0.125	0
A4.txt	150	torneo	0.9	0.15	0
A4.txt	150	torneo	0.95	0.01	10
A4.txt	150	torneo	0.95	0.05	0
A4.txt	150	torneo	0.95	0.1	0
A4.txt	150	torneo	0.95	0.125	0
A4.txt	150	torneo	0.95	0.15	2

Sudoku	Tam. pobl	Selector	pcruce	pmut	fitness
A5.txt	100	ruleta	0.8	0.01	2
A5.txt	100	ruleta	0.8	0.05	2
A5.txt	100	ruleta	0.8	0.1	0
A5.txt	100	ruleta	0.8	0.125	6
A5.txt	100	ruleta	0.8	0.15	6
A5.txt	100	ruleta	0.85	0.01	4
A5.txt	100	ruleta	0.85	0.05	0
A5.txt	100	ruleta	0.85	0.1	2
A5.txt	100	ruleta	0.85	0.125	5
A5.txt	100	ruleta	0.85	0.15	3
A5.txt	100	ruleta	0.9	0.01	4
A5.txt	100	ruleta	0.9	0.05	2
A5.txt	100	ruleta	0.9	0.1	2
A5.txt	100	ruleta	0.9	0.125	2
A5.txt	100	ruleta	0.9	0.15	4
A5.txt	100	ruleta	0.95	0.01	0
A5.txt	100	ruleta	0.95	0.05	2
A5.txt	100	ruleta	0.95	0.1	2
A5.txt	100	ruleta	0.95	0.125	2
A5.txt	100	ruleta	0.95	0.15	7
A5.txt	150	ruleta	0.8	0.01	4
A5.txt	150	ruleta	0.8	0.05	0
A5.txt	150	ruleta	0.8	0.1	4
A5.txt	150	ruleta	0.8	0.125	5
A5.txt	150	ruleta	0.8	0.15	0
A5.txt	150	ruleta	0.85	0.01	4
A5.txt	150	ruleta	0.85	0.05	4
A5.txt	150	ruleta	0.85	0.1	2
A5.txt	150	ruleta	0.85	0.125	2
A5.txt	150	ruleta	0.85	0.15	2
A5.txt	150	ruleta	0.9	0.01	2
A5.txt	150	ruleta	0.9	0.05	2
A5.txt	150	ruleta	0.9	0.1	2
A5.txt	150	ruleta	0.9	0.125	4
A5.txt	150	ruleta	0.9	0.15	2
A5.txt	150	ruleta	0.95	0.01	3
A5.txt	150	ruleta	0.95	0.05	2
A5.txt	150	ruleta	0.95	0.1	2
A5.txt	150	ruleta	0.95	0.125	2
A5.txt	150	ruleta	0.95	0.15	0

Sudoku	Tam. pobl	Selector	pcruce	pmut	fitness
A5.txt	100	torneo	0.8	0.01	2
A5.txt	100	torneo	0.8	0.05	2
A5.txt	100	torneo	0.8	0.1	0
A5.txt	100	torneo	0.8	0.125	0
A5.txt	100	torneo	0.8	0.15	0
A5.txt	100	torneo	0.85	0.01	5
A5.txt	100	torneo	0.85	0.05	0
A5.txt	100	torneo	0.85	0.1	0
A5.txt	100	torneo	0.85	0.125	0
A5.txt	100	torneo	0.85	0.15	2
A5.txt	100	torneo	0.9	0.01	5
A5.txt	100	torneo	0.9	0.05	2
A5.txt	100	torneo	0.9	0.1	2
A5.txt	100	torneo	0.9	0.125	2
A5.txt	100	torneo	0.9	0.15	0
A5.txt	100	torneo	0.95	0.01	4
A5.txt	100	torneo	0.95	0.05	0
A5.txt	100	torneo	0.95	0.1	0
A5.txt	100	torneo	0.95	0.125	2
A5.txt	100	torneo	0.95	0.15	0
A5.txt	150	torneo	0.8	0.01	7
A5.txt	150	torneo	0.8	0.05	0
A5.txt	150	torneo	0.8	0.1	0
A5.txt	150	torneo	0.8	0.125	0
A5.txt	150	torneo	0.8	0.15	0
A5.txt	150	torneo	0.85	0.01	2
A5.txt	150	torneo	0.85	0.05	0
A5.txt	150	torneo	0.85	0.1	0
A5.txt	150	torneo	0.85	0.125	0
A5.txt	150	torneo	0.85	0.15	0
A5.txt	150	torneo	0.9	0.01	6
A5.txt	150	torneo	0.9	0.05	0
A5.txt	150	torneo	0.9	0.1	0
A5.txt	150	torneo	0.9	0.125	0
A5.txt	150	torneo	0.9	0.15	0
A5.txt	150	torneo	0.95	0.01	4
A5.txt	150	torneo	0.95	0.05	0
A5.txt	150	torneo	0.95	0.1	0
A5.txt	150	torneo	0.95	0.125	0
	150	torneo	0.95	0.15	0

Análisis de las pruebas de ajuste

Para 12000 generaciones y para los sudokus de pruebas A1, A2, A3, A4 y A5, tenemos que:

- El método de selección mejor es torneo (GATournamentSelector), ya que podemos observar en las tablas de valores que con el selector ruleta (GARouletteWheelSelector), nunca se consigue que para una misma configuración obtengamos un resultado de fitness igual a 0 para los cinco sudokus. Sin embargo, cuando se usa torneo, esto se da con varias configuraciones.
- Para tamaño de población 150 y basándonos en las configuraciones que nos dan un resultado de fitness igual a 0 para todos los casos de prueba, vemos que independientemente de la probabilidad de cruce, una probabilidad de mutación de 0.1 o 0.125 nos asegura que vamos a obtener una solución al sudoku presentado.
- Por otro lado, respecto a la población de 100 individuos es difícil sacar una conclusión clara debido a que tan solo obtenemos 4 configuraciones exitosas, en las que no encontramos ningún nexo de unión.

Manual y asignación

Por tanto, fijándonos en los resultados obtenidos en el apartado anterior podemos decir que los parámetros están claramente definidos. La configuración que mejores resultados dará es la siguiente:

- Selector: torneo (GATournamentSelector).
- Tamaño de la población: 150.
- Probabilidad de mutación: 0.1 o 0.125.
- Probabilidad de cruce: indiferente, usaremos 0.8.

Junto con el ejecutable del programa y en referencia al posible uso de un usuario, se adjunta un manual para la ejecución.

Protocolo de resolución de los casos de usuario

Usando los parámetros que hemos concluido son los más adecuados para nuestro problema vamos a aplicarlos a la resolución de los tres casos del usuario. Como podemos comprobar, usando los parámetros que hemos concluido son los mejores, obtenemos la solución para cada uno de los casos de usuario.

1.txt:

Solución del sudoku con los siguientes parámetros:

1.txt 150 torneo 0.8 0.1

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

Mejor valor fitness es 0

Solución del sudoku con los siguientes parámetros:

1.txt 150 torneo 0.8 0.125

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

Mejor valor fitness es 0

2.txt:

Solución del sudoku con los siguientes parámetros:

2.txt 150 torneo 0.8 0.1

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

Mejor valor fitness es 0

Solución del sudoku con los siguientes parámetros:

2.txt 150 torneo 0.8 0.125

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

Mejor valor fitness es 0

3.txt:

Solución del sudoku con los siguientes parámetros:

3.txt 150 torneo 0.8 0.1

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

Mejor valor fitness es 0

Solución del sudoku con los siguientes parámetros:

3.txt 150 torneo 0.8 0.125

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

Mejor valor fitness es 0

Bibliografía

- [1] Genetics Algorithms (2006). http://www.bluetronix.net/Genetic_Algorithms.htm. Imagen de portada.
- [2] Matthew Wall (1996). GAlib: A C++ Library of Genetic Algorithm Components. Documentación de la librería GAlib, usada para algunas definiciones teóricas y para la revisión del código interno del algoritmo. Además, con ayuda del código de algunas funciones que aparecen, se pudo realizar los apartados del casting.
- [3] Sistemas Inteligentes (2016-2017). Fundamentos teóricos. Diapositivas de la asignatura de Sistemas Inteligentes, 3º Ingeniera Informática en la UMU. Usadas para responder a algunas cuestiones teóricas.
- [4] Sistemas Inteligentes (2016-2017). Fundamentos prácticos. Diapositivas de la asignatura de Sistemas Inteligentes, 3º Ingeniera Informática en la UMU. Usadas para entender cómo funciona el algoritmo y cómo interpretar los resultados.