

¿CÓMO RESOLVER UN SUDOKU CON UN ALGORITMO GENÉTICO?



RAUL EXPOSITO
COM

19 de septiembre de 2009

INDICE

| | |
|--|----|
| 1. Introducción..... | 03 |
| 2. ¿Qué son los algoritmos genéticos?..... | 03 |
| 3. ¿Cómo encajar todo esto? | 05 |
| 4. Problema a resolver..... | 07 |
| 5. Tecnologías utilizadas..... | 08 |
| 6. Resolución del problema..... | 09 |
| 7. Conclusiones..... | 14 |
| 8. Referencias..... | 14 |

1. Introducción

A lo largo de este documento voy a explicar cómo he conseguido resolver un sudoku de 3x3 utilizando algoritmos genéticos.

¿Es esta técnica adecuada para resolver un problema de este tipo? A priori no lo es, y adelantándome a las conclusiones, diré que a posteriori tampoco. Hay algoritmos concretos creados única y específicamente para resolver sudokus que lo hacen en menos tiempo, consumiendo menos memoria y menos CPU.

Entonces, ¿por qué utilizar un algoritmo genético en este problema?. Pues básicamente porque me pareció un ejemplo práctico y sencillo, ya que casi todo el mundo sabe lo que es un sudoku y, de este modo, es mucho más fácil transmitir qué es lo que hace el algoritmo genético sin entrar en *aburridas* cuestiones teóricas o técnicas.

Además es un buen ejemplo porque, cuando lo estaba programando, me encontré con alguna dificultad que no pensé que fuera a encontrar y que lo hace más interesante. En la práctica, resolver un sudoku con esta técnica es algo más complicado que en la teoría, y es que ya dicen que:

“En teoria, la teoría y la práctica son lo mismo, pero en la práctica, no lo son”

Espero que, cuando termines de leer este texto, hayas encontrado este experimento interesante y que lo haya explicado correctamente para haberte facilitado el aprender algo nuevo.

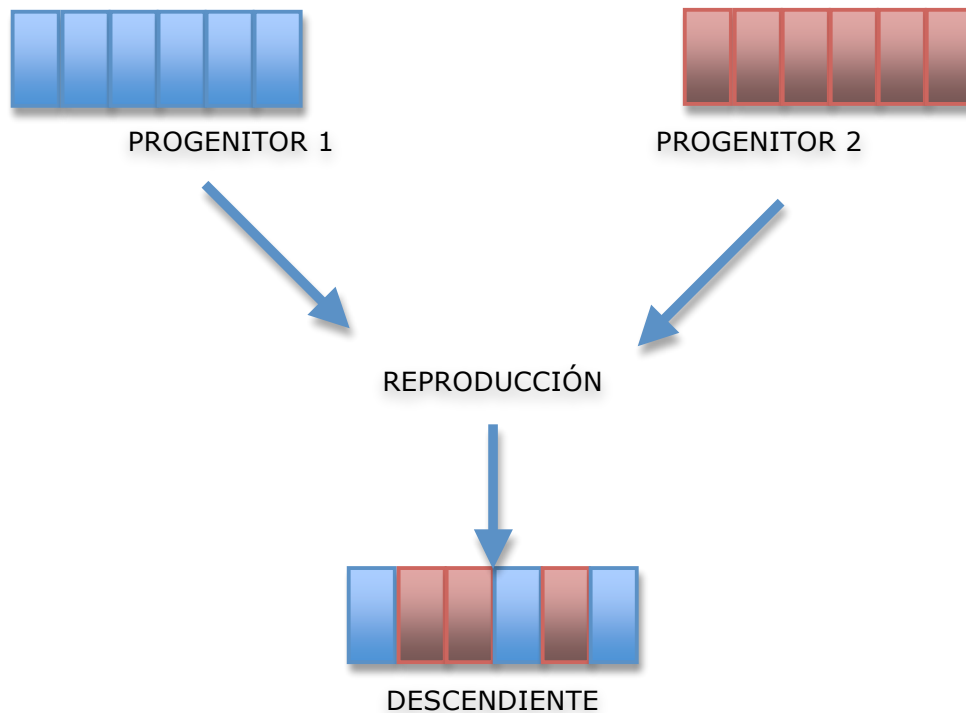
2. ¿Qué son los algoritmos genéticos?

No voy a entrar mucho en detalle acerca de qué son los algoritmos genéticos, ya que por mucho y muy bien que diga siempre diré menos y peor que otros que saben más que yo, como Juan Julián Merelo Guervós [1], Adam Marczyk [2], el autor de este texto [3] o la wikipedia [4].

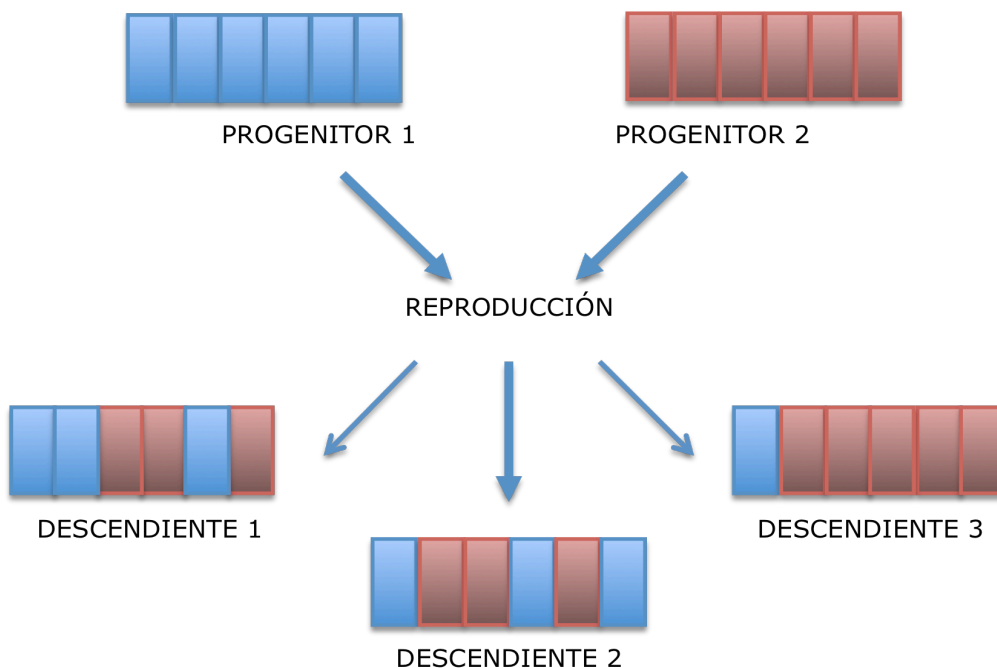
Digamos que los algoritmos genéticos basan su funcionamiento en la teoría de la evolución propuesta por Darwin. Según esta teoría, sólo aquellos individuos que **estén mejor adaptados al medio** se reproducen, mientras que los demás mueren sin descendencia. De este modo se consigue que a lo largo de las sucesivas generaciones los individuos estén cada vez mejor adaptados al medio en el que habitan y, en definitiva, que la especie evolucione.

Cada individuo, como sabemos, **posee un código genético** que se cruza con el de otro individuo durante la reproducción, proceso tras el cual se obtiene un código genético producto de la mezcla entre los códigos genéticos de los progenitores.

Dicho código genético definirá a un nuevo individuo descendiente de los anteriores.



Aunque en este ejemplo se genera un solo descendiente tras el proceso reproductivo, esto no tiene por qué ser siempre así, pudiendo de una vez ser generados varios descendientes que pueden tener códigos genéticos diferentes.



Como producto derivado de la reproducción aparece el concepto de **generación**. En los dibujos anteriores los progenitores serían la generación 1 mientras que los descendientes serían la generación 2. Llegado el momento, los miembros de la generación 1 desaparecerán, los miembros de la generación 2 se reproducirán, asumiendo el papel de progenitores, y generarán a los nuevos descendientes que formarán una nueva generación 3, y así sucesivamente.

3. ¿Cómo encajar todo esto?

Como veis, todo lo comentado hasta ahora está basado en el funcionamiento de la naturaleza. Sin embargo, ¿cómo podemos adaptar este mecanismo para que nos pueda ser de ayuda para resolver sudokus?

En el apartado anterior he señalado algunas cosas clave en negrita, que son las que necesitaremos para poder hacer funcionar el algoritmo:

- **Un medio.** Necesitamos *algo* donde los individuos se puedan poner a prueba, donde puedan competir entre ellos, donde puedan demostrar cómo de adaptados están. Algo que los *presione* a evolucionar.

En la Naturaleza esta definición es muy amplia. El medio podría ser una estepa, un desierto, un charco de agua, una fosa submarina, un poco de luz, una caverna, la selva amazónica, etc. Viene a ser un lugar o un espacio compartido en el cual los individuos deben luchar por los recursos, de tal modo que los mejor adaptados a ese medio puedan reproducirse.

En nuestro caso, *¿cuál crees que debería ser el medio en el cual deberíamos poder ver cómo de adaptados están los individuos?*

- **Individuos.** Tiene que haber *entidades* que luchen entre sí y se desarrollen en el medio, tratando de adaptarse cada vez más a él a lo largo de las generaciones. Cuanto más adaptado esté el individuo al medio, más fácilmente podrá reproducirse.

En este problema, *¿quiénes deberían ser los individuos?*

- **Un código genético.** El cual contendrá toda la información que *definirá* al individuo. Si el individuo que se obtiene del código genético tiene éxito en el medio, conseguirá que sus genes puedan mezclarse con los de otro individuo también exitoso y que estos sirvan de base en el código genético de un nuevo individuo de la siguiente generación. En cambio, si el individuo no tiene éxito, no podrá reproducirse y, por tanto, sus genes no se incluirán en la siguiente generación.

Para vosotros, *¿cuál es el código genético?*

¿De verdad quieres pasar la hoja sin haberlo pensado siquiera?, ¿no quieres intentarlo? Son sólo tres preguntitas de nada. Venga anda, ánimo 😊

Empecemos a responder la segunda pregunta. Si estamos tratando de resolver un tablero de sudoku incompleto, **los individuos serán los sudokus completos, sean válidos o no, que traten de resolver el problema planteado en el tablero.**

¿Qué un individuo tiene todo unos en una fila?, no importa. ¿Qué tiene valores repetidos en las columnas o en los cuadrados interiores?, da igual, lo importante es que esté intentando resolver el tablero de sudoku. El fin último es dar con un sudoku que resuelva el problema, y mientras tanto no importa que los individuos sean sudokus que no cumplen las reglas y, por tanto, no son válidos.

Una vez sabemos quienes son los individuos podemos saber fácilmente quién es **el medio**, y es que los sudokus tienen un objetivo en mente: resolver **el problema del sudoku incompleto**. Es problema el escenario en el que se enfrentan los sudokus, ya que aquellos que estén más cerca de resolverlo serán aquellos que estén más adaptados al medio, mientras que los que están más lejos de resolverlo, los que menos.

¿Se reproducirán, por tanto, los sudokus que estén más cerca de resolver el tablero de sudoku incompleto con más facilidad?. Pues así es, y aunque suene difícil de creer y haya que echar algo de imaginación, los sudokus se reproducirán y tendrán descendientes sudokus.

Y finalmente, ¿cuál será el **código genético** de los sudokus?, pues el **valor que cada uno de ellos tenga en las casillas**. Por ejemplo, para el siguiente sudoku:

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

Este sería su código genético:

| |
|---|
| 136428957459176832287395641693581724845267193721943568568734219914852376372619485 |
|---|

En función a este código genético se generará un individuo sudoku y, en función a cómo cumpla el individuo con el tablero del sudoku incompleto, le será más fácil o no reproducirse.

4. Problema a resolver

Lo escrito hasta ahora nos da una visión global del mecanismo que rige el funcionamiento del algoritmo genético. Sabemos que hay unos individuos, los cuales están formados por un código genético que se cruza y entremezcla con el código genético de otros individuos para formar nuevos individuos, y también sabemos que hay un medio en el cual los individuos luchan por poder reproducirse.

También sabemos que a lo largo de las generaciones los individuos estarán más y más adaptados al medio, ya que los sudokus con éxito se reproducirán, con lo que los individuos descendientes tendrán los genes de los progenitores y, posiblemente con ellos, las características que hayan hecho que éstos tengan éxito en el medio.

Sin embargo, todo esto de momento carece de insercia: nos falta una pieza importante que haga que todo esto tenga sentido. Algo que en todo el texto anterior está implícito pero no hemos señalar de manera implícita: falta definir **qué es lo que hace que un individuo sea exitoso en el medio**.

Esta manera de medir cómo de bueno o malo es un individuo en un medio determinado se denomina *fitness*, afinidad o idoneidad. Aunque habitualmente se utilice el anglicismo, nos referiremos a este término en este documento como **idoneidad**.

¿Y cuál es, en concreto, la función de la idoneidad en el desarrollo de un algoritmo genético? Su papel es muy sencillo: **evaluar a los individuos** utilizando, para ello, una **función de idoneidad**.

Imaginemos que la idoneidad es una caja con una pantalla y con una entrada por la cual metemos a un individuo seleccionado al azar. Tras esperar un rato a que la caja evalúe al individuo, ésta nos devuelve un resultado en pantalla. ¿Quién determina cuál es el resultado que aparece en pantalla?, la función de idoneidad, ¿y qué nos dice el resultado de la pantalla?, pues cómo de bueno o malo es el individuo para la función de idoneidad.

Pongamos un caso. Tenemos una población con cuatro sudokus, y uno a uno los introducimos en nuestra caja que calcula la idoneidad. Tras evaluarlos, obtenemos lo siguiente:

- Sudoku 1: MALO
- Sudoku 2: BUENO
- Sudoku 3: MUY BUENO
- Sudoku 4: REGULAR

Lo que haremos será facilitar que se reproduzcan los sudokus 2 y 3, sin evitar que los demás también se reproduzcan, intentando que las siguientes generaciones posean aquellos genes que consiguen que estos dos sudokus tengan características que los hacen exitosos en el medio.

Como podeis ver, la función de idoneidad está **muy relacionada** con el medio, de tal modo que los individuos que no estén adaptados al medio deben obtener calificaciones bajas, dejando las calificaciones altas para los individuos muy adaptados.

Si nuestra función de idoneidad no es la adecuada es posible que nuestros individuos nunca se adapten al medio. Esto, aplicado a los algoritmos genéticos, significa que **si la función de idoneidad no es buena no será posible encontrar soluciones**.

En el caso que nos ocupa, ¿cómo haremos para medir cómo de bueno o malo es un individuo? Un sudoku es un cuadrado de 9x9 casillas, por lo que tiene un total de 81 números que deben ser correctamente colocados. Lo que haremos será evaluar a los sudokus uno a uno utilizando nuestra función de idoneidad, siguiendo para ello las reglas del sudoku:

- **REGLA 1:** no debe haber valores repetidos en las filas.

Si tenemos la siguiente fila del sudoku:

| | | | | | | | | |
|---|----------|---|----------|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 3 | 4 | 5 | 6 | 7 |
|---|----------|---|----------|---|---|---|---|---|

Podemos decir que tiene 7 valores únicos (ya que el 1 y el 3 están repetidos), con lo que el valor de esta fila será 7. Hay que repetir este experimento en las otras 8 filas para saber el número de valores correctos en las filas.

- **REGLA 2:** no debe haber valores repetidos en columnas.

De un modo similar al anterior, una a una se evalúan las columnas con el objetivo de determinar cuantos valores únicos contiene cada una de ellas.

- **REGLA 3:** no debe haber valores repetidos en cada columna de 3x3.

Igual que en los casos anteriores, se van seleccionando los cuadrados de 3x3 del sudoku y se determina cuántos valores únicos poseen.

Y ahora, ¿cómo combinamos los resultados que nos devuelven las tres reglas? Lo veremos más adelante. De todos modos hay que tener en cuenta que es posible que, por ejemplo, todas las filas tengan valores únicos pero las columnas o los cuadrados de 3x3 no. El sudoku estará completo cuando en las filas, en las columnas y en los cuadrados de 3x3 no haya valores repetidos.

5. Tecnologías utilizadas

El software utilizado para poder implementar este problema ha sido el siguiente:

- JGAP 3.4.3 como motor del algoritmo genético.
- Java 6 como lenguaje de programación.
- JUnit 4 como base para la realización de pruebas.

6. Resolución del problema

El sudoku que vamos a resolver con el algoritmo genético es el siguiente:

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

Como veis aparentemente es muy sencillo, ya que incluso tiene un cuadrado de 3x3 resuelto. Para nosotros resolverlo sería muy fácil, ya que mentalmente utilizamos reglas que permiten rellenar los huecos uno a uno, o incluso podríamos programar un algoritmo específico que siguiese esas reglas por nosotros.

Pero el algoritmo genético no funciona así. No utiliza reglas, tan sólo rellena huecos con números sin mirar si en esa fila o columna ese número ya existía y se le evalúa en base a si sus números se repiten o no en filas, columnas y cuadrados de 3x3.

Este sudoku en particular tiene 31 huecos que pueden estar rellenos por números del 1 al 9, lo que significa que el número de posibilidades es de $9 \times 9 \times 9 \times 9 \dots$ y así 31 veces. Echando mano de nuestra amiga calculadora, ésta nos dice que hay:

$$9^{31} = 3,82 * 10^{29} =$$

38200000000000000000000000000000

¡¡posibles combinaciones!!

Como veis, es muy importante elegir el tipo de algoritmo que vais a utilizar a la hora de resolver un problema. Un algoritmo específico basado en reglas quizá encuentre algunos cientos, miles o en el peor de los casos millones de

combinaciones que evaluar. El algoritmo genético, como veis, debe buscar entre cantidades bastante superiores.

Pero, ¿eso significa que el algoritmo genético debe evaluar todas y cada una de las combinaciones?. **En absoluto**, de hecho el algoritmo genético evaluará tan sólo a poblaciones de 50 en 50 individuos del modo que sigue:

1. Al principio generará 50 individuos rellenando los huecos del sudoku a resolver aleatoriamente. A estos 50 individuos se les denominará **generación 1**.

Mide la afinidad de esos 50 individuos. Los 5 mejores individuos se copian, y se generan otros 45 reproduciendo entre sí a los sudokus de la generación 1, haciendo que sea más fácil que se reproduzcan los que tienen más números bien colocados.

2. A estos 5+45 individuos producidos anteriormente se les denomina **generación 2**. Los 50 individuos de la generación 1 desaparecen.

De nuevo se mide la afinidad de los 50 individuos, generando 45 nuevos individuos por la reproducción entre ellos , y copiando a los 5 con mayor afinidad.

3. A estos 5+45 individuos se les denomina **generación 3**, los 50 individuos de la generación 2 desaparecen y repite el mecanismo de copia y reproducción.

¿Y cuando pararemos de crear generaciones?, cuando lleguemos a un número máximo de poblaciones o cuando encontremos una solución al problema.

En nuestro caso, se llegó a la solución en la generación 2845, donde $2845 * 50$ son *sólo* 142250 evaluaciones, que es un número bastante inferior a la barbaridad de posibilidades que teníamos inicialmente.

El algoritmo genético, en el fondo, lo que hace es guiar a la aleatoriedad. Parte de 50 puntos aleatorios de entre tantas posibilidades y se acerca a lo largo de las generaciones a una solución gracias a una brújula tan precisa como adecuada sea la función de afinidad.

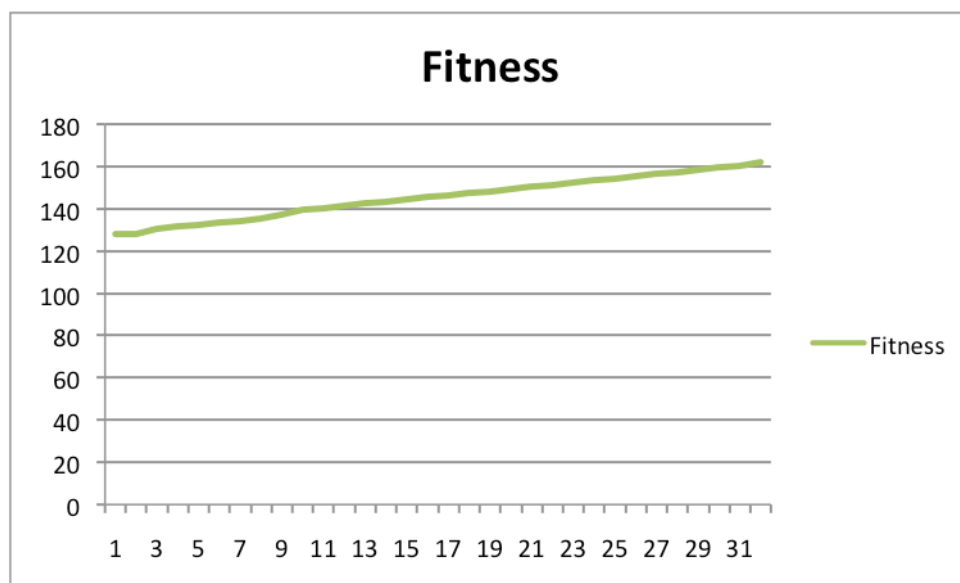
Como ya mencionamos anteriormente, la función de afinidad es **clave** en la resolución de los problemas mediante el algoritmo genético, ya que según como la definamos podremos o no encontrar soluciones. Para poder resolver este problema he tenido que probar distintas funciones de afinidad partiendo de las 3 reglas mencionadas en la página 8, y puedo sacar las siguientes conclusiones:

- Si evalúas a los individuos contando la cantidad de números bien colocados en filas, columnas y cuadrados de 3x3 verás que el algoritmo genético tiene mucha tendencia a rellenar correctamente los cuadrados de 3x3 y, una vez que lo hace, opone mucha resistencia a que se rellenen correctamente las filas y las columnas, con lo que no se llega a ninguna solución.
- Si evalúas a los individuos sumando solamente los valores bien colocados en las filas y en los cuadrados de 3x3 verás que el algoritmo genético coloca los valores de las filas con mucha facilidad, le cuesta mucho rellenar a los cuadrados de 3x3 y no es capaz de colocar columnas.
- Si multiplicas el número de valores bien colocados en filas por dos, en columnas por dos, y en cuadrados de 3x3 por uno, verás que no es capaz de colocar números y evolucionar en ninguno de los 3.

Y es que si lo piensas, las filas, columnas y cuadrados mantienen internamente una lucha por los números. Los cuadrados lucharán por mantener los valores bien colocados dentro de ellos, al igual que las filas y las columnas. Es más, si consigues colocar correctamente los valores de dos de ellos habrás conseguido resolver el sudoku.

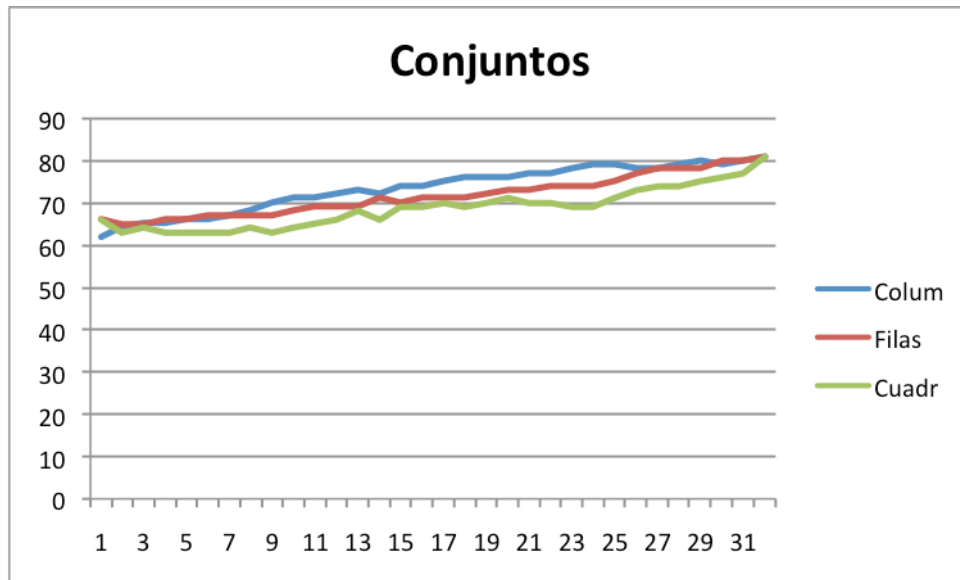
Es por ello que la función de afinidad que finalmente he utilizado ha sido muy sencilla: tan sólo **contar el número de elementos bien colocados en filas y en columnas**.

Como hay 81 números en filas y 81 números en columnas (donde $81 = 9 \times 9$), el valor máximo de números bien colocados siguiendo esta estrategia es 162, con lo que cuando consigamos 162 números bien colocados habremos resuelto el sudoku. Esta gráfica representa cómo ha ido evolucionando la cantidad de números bien colocados a lo largo de las generaciones, mostrándose sólo las 31 generaciones donde ha habido mejoras.



Donde *fitness* representa el valor de la afinidad, y como veis ha ido mejorando progresivamente a lo largo de las generaciones.

También es posible ver cómo se han ido colocando los valores en filas y en columnas:



Como se puede ver, algunas veces las columnas tienen los números mejor ordenados, otras veces son las filas las que ordenan mejor los números y también se puede ver cómo, a medida que ambas van mejorando, se va ordenando correctamente el contenido de los cuadrados de 3x3.

Las gráficas se han obtenido con los siguientes datos, obtenidos de la ejecución del programa:

| | | | | | |
|------|-------------------------|--------------|---------|---------|---------|
| INFO | Main - Generacion: 0000 | fitness: 128 | col: 62 | fil: 66 | cua: 66 |
| INFO | Main - Generacion: 0002 | fitness: 129 | col: 64 | fil: 65 | cua: 63 |
| INFO | Main - Generacion: 0003 | fitness: 130 | col: 65 | fil: 65 | cua: 64 |
| INFO | Main - Generacion: 0004 | fitness: 131 | col: 65 | fil: 66 | cua: 63 |
| INFO | Main - Generacion: 0005 | fitness: 132 | col: 66 | fil: 66 | cua: 63 |
| INFO | Main - Generacion: 0006 | fitness: 133 | col: 66 | fil: 67 | cua: 63 |
| INFO | Main - Generacion: 0008 | fitness: 134 | col: 67 | fil: 67 | cua: 63 |
| INFO | Main - Generacion: 0009 | fitness: 135 | col: 68 | fil: 67 | cua: 64 |
| INFO | Main - Generacion: 0012 | fitness: 137 | col: 70 | fil: 67 | cua: 63 |
| INFO | Main - Generacion: 0016 | fitness: 139 | col: 71 | fil: 68 | cua: 64 |
| INFO | Main - Generacion: 0022 | fitness: 140 | col: 71 | fil: 69 | cua: 65 |
| INFO | Main - Generacion: 0023 | fitness: 141 | col: 72 | fil: 69 | cua: 66 |
| INFO | Main - Generacion: 0026 | fitness: 142 | col: 73 | fil: 69 | cua: 68 |
| INFO | Main - Generacion: 0029 | fitness: 143 | col: 72 | fil: 71 | cua: 66 |

| | | | | | |
|------|-------------------------|--------------|---------|---------|---------|
| INFO | Main - Generacion: 0030 | fitness: 144 | col: 74 | fil: 70 | cua: 69 |
| INFO | Main - Generacion: 0034 | fitness: 145 | col: 74 | fil: 71 | cua: 69 |
| INFO | Main - Generacion: 0035 | fitness: 146 | col: 75 | fil: 71 | cua: 70 |
| INFO | Main - Generacion: 0041 | fitness: 147 | col: 76 | fil: 71 | cua: 69 |
| INFO | Main - Generacion: 0045 | fitness: 148 | col: 76 | fil: 72 | cua: 70 |
| INFO | Main - Generacion: 0050 | fitness: 149 | col: 76 | fil: 73 | cua: 71 |
| INFO | Main - Generacion: 0052 | fitness: 150 | col: 77 | fil: 73 | cua: 70 |
| INFO | Main - Generacion: 0067 | fitness: 151 | col: 77 | fil: 74 | cua: 70 |
| INFO | Main - Generacion: 0101 | fitness: 152 | col: 78 | fil: 74 | cua: 69 |
| INFO | Main - Generacion: 0106 | fitness: 153 | col: 79 | fil: 74 | cua: 69 |
| INFO | Main - Generacion: 0141 | fitness: 154 | col: 79 | fil: 75 | cua: 71 |
| INFO | Main - Generacion: 0199 | fitness: 155 | col: 78 | fil: 77 | cua: 73 |
| INFO | Main - Generacion: 0202 | fitness: 156 | col: 78 | fil: 78 | cua: 74 |
| INFO | Main - Generacion: 0210 | fitness: 157 | col: 79 | fil: 78 | cua: 74 |
| INFO | Main - Generacion: 0212 | fitness: 158 | col: 80 | fil: 78 | cua: 75 |
| INFO | Main - Generacion: 0256 | fitness: 159 | col: 79 | fil: 80 | cua: 76 |
| INFO | Main - Generacion: 0270 | fitness: 160 | col: 80 | fil: 80 | cua: 77 |
| INFO | Main - Generacion: 2845 | fitness: 162 | col: 81 | fil: 81 | cua: 81 |

INFO Main -

1 3 6 4 2 8 9 5 7

4 5 9 1 7 6 8 3 2

2 8 7 3 9 5 6 4 1

6 9 3 5 8 1 7 2 4

8 4 5 2 6 7 1 9 3

7 2 1 9 4 3 5 6 8

5 6 8 7 3 4 2 1 9

9 1 4 8 5 2 3 7 6

3 7 2 6 1 9 4 8 5

Como se puede ver en la generación 270 se consigue un individuo muy bueno, el cual no se mejora hasta la generación 2845.

7. Conclusiones

¿Es este tipo de algoritmos adecuado para resolver este tipo de problemas? Como habéis podido comprobar no, no lo es, ya que la cantidad de posibilidades entre las cuales debe encontrar una solución a un sudoku sencillito es inmensa, y crece exponencialmente cada vez que le quitamos un número.

Aparte, para solucionar este problema tan sencillo ha sido necesario evaluar 142250 posibles soluciones, cuando un algoritmo específico seguramente habría evaluado tan solo algunos cientos de ellas.

¿Cuándo deberemos entonces utilizar un algoritmo genético?. Cuando intentemos resolver un problema para el cual no somos capaces de generar un algoritmo específico por nosotros mismos debido a su complejidad. En ese caso deberemos estar muy atentos a los resultados que generación a generación vamos obteniendo para saber si la brújula que guía la evolución, esto es, la función de afinidad, nos lleva donde queremos ir.

8. Referencias

[1] Informática evolutiva: algoritmos genéticos
<http://geneura.ugr.es/~jmerelo/ie/ags.htm>

[2] Algoritmos genéticos y computación evolutiva
<http://the-geek.org/docs/algen/>

[3] Algoritmos genéticos
<http://eddyalfaro.galeon.com/geneticos.html>

[4] Algoritmo genético
http://es.wikipedia.org/wiki/Algoritmo_genetico