

ALGORITMOS GENÉTICOS Y EVOLUTIVOS

PRÁCTICA 1

(Optimización de Sensores en Smart Cities)



Grado en Ingeniería Informática

Campus de Colmenarejo

Curso 2020/2021

Autor

Eduardo Ureña Toledano - 100329937@alumnos.uc3m.es

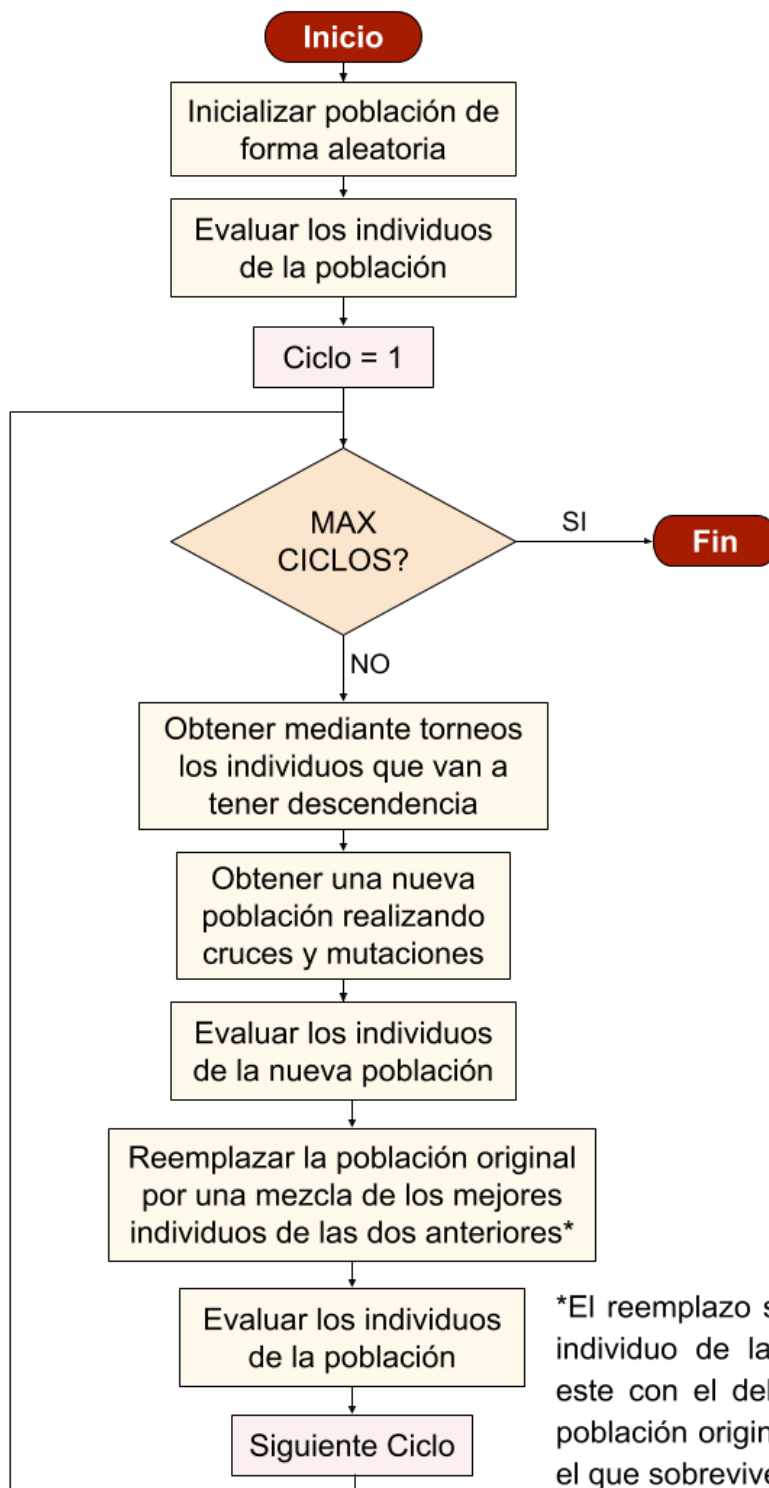
Índice

1. Introducción	3
2. Diagrama del Algoritmo Genético	3
3. Parámetros propuestos y análisis comparativo de los resultados	4
3.1. Primera situación (4 estaciones)	4
3.2. Segunda situación (24 estaciones)	7
4. Conclusiones y problemas encontrados	9

1. Introducción

En este documento se expondrán los resultados obtenidos de un Algoritmo Genético, desarrollado para minimizar el coste que supone tener estaciones que midan la calidad del aire de Madrid, por medio de sensores, pero perdiendo la menor precisión posible.

2. Diagrama del Algoritmo Genético



3. Parámetros propuestos y análisis comparativo de los resultados

He probado el Algoritmo Genético suponiendo dos situaciones diferentes:

1) Tan sólo hay 4 estaciones disponibles de las 24 posibles, lo que indica que cada individuo (cada solución) estará compuesto por un cromosoma de 64 genes (una cadena de 64 bits) teniendo en cuenta que cada estación consta de 16 sensores.

Para esta situación el fitness de cada individuo se obtendrá llamando al siguiente servicio web: http://memento.evannai.inf.uc3m.es/age/test?c=<cromosoma_64_bits>

2) Las 24 estaciones posibles están disponibles, lo que indica que cada individuo (cada solución) estará compuesto por un cromosoma de 384 genes (una cadena de 384 bits).

Para esta situación el fitness de cada individuo se obtendrá llamando al siguiente servicio web: http://memento.evannai.inf.uc3m.es/age/alfa?c=<cromosoma_384_bits>

3.1. Primera situación (4 estaciones)

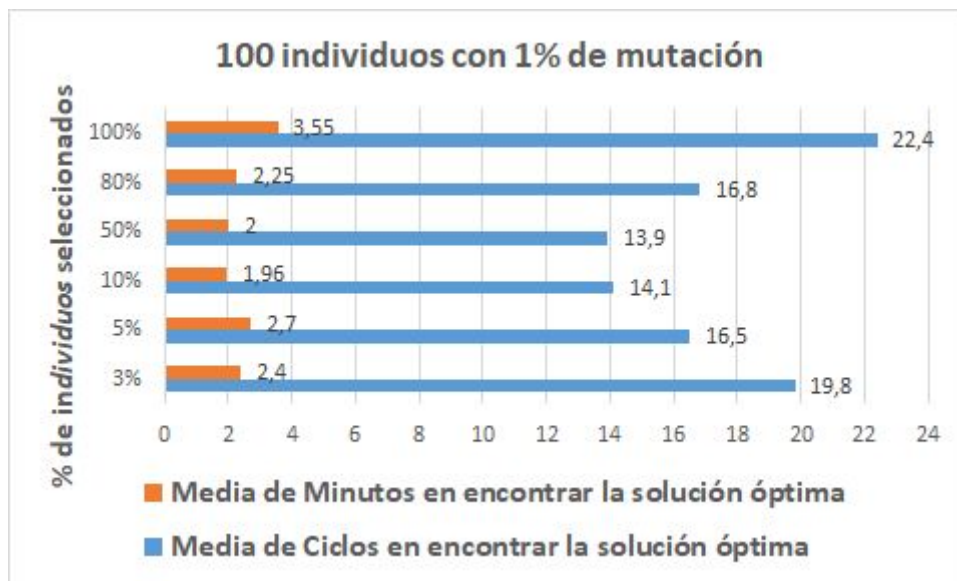
Para esta primera situación he ejecutado 10 veces cada combinación de parámetros, y además he decidido usar una población de 100 individuos.

He empezado con un porcentaje de mutación de un 1% y teniendo en cuenta que el porcentaje más adecuado de individuos a escoger aleatoriamente en los torneos suele ser de entre el 3% y el 5%, he elegido estos dos para empezar, obteniendo una media de 19.8 \approx 20 ciclos en encontrar la solución óptima (0.0) escogiendo un 3%, y una media de 16.5 \approx 17 ciclos escogiendo un 5%.

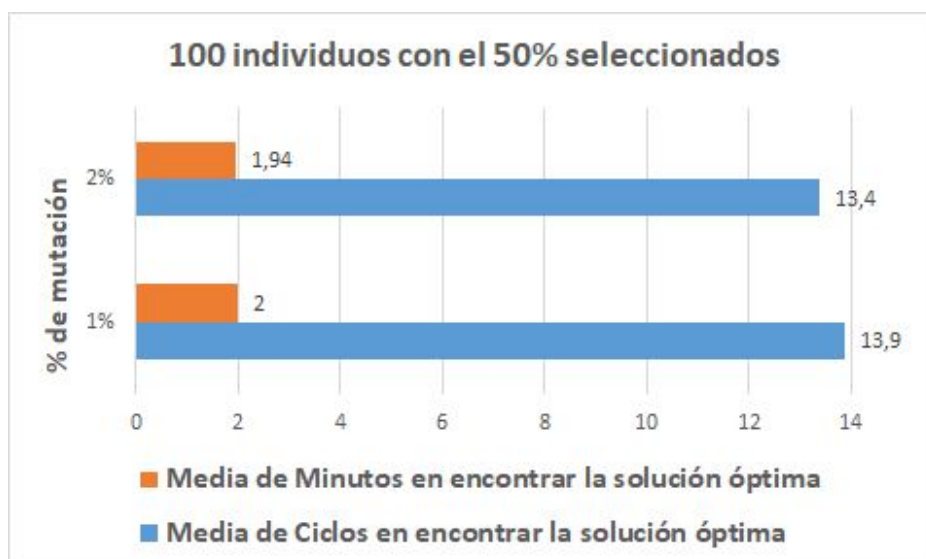
Al ver que al aumentar el porcentaje disminuye la media de ciclos en encontrar la mejor solución, he decidido aumentarlo hasta el 10%, obteniendo una media de 14.1 \approx 14 ciclos en encontrar la solución, que al seguir disminuyendo me lleva a probar con un 50%, consiguiendo encontrar la solución óptima en una media de 13.9 \approx 14 ciclos, que aunque por poco sigue siendo una mejor opción en el problema que nos ocupa.

Por último, antes de cambiar el porcentaje de mutación, aunque suponiendo que en este caso la regla de aumentar el porcentaje no iba a seguir funcionando, pruebo con un 80%, obteniendo esta vez la mejor solución en una media de 16.8 \approx 17 ciclos, indicando así que esta combinación de valores es peor, pero no mucho peor que el 5% y mejor que el 3%, por lo que pruebo cogiendo un 100% de los individuos, obteniendo así tan solo el mejor individuo mutado varias veces en cada ciclo al servir el cruce en este caso tan solo para tener más ejemplares del mismo individuo. Obtengo la solución óptima en una media de 22.4 \approx 22 ciclos, lo que me indica que es la peor opción hasta el momento, pero al no

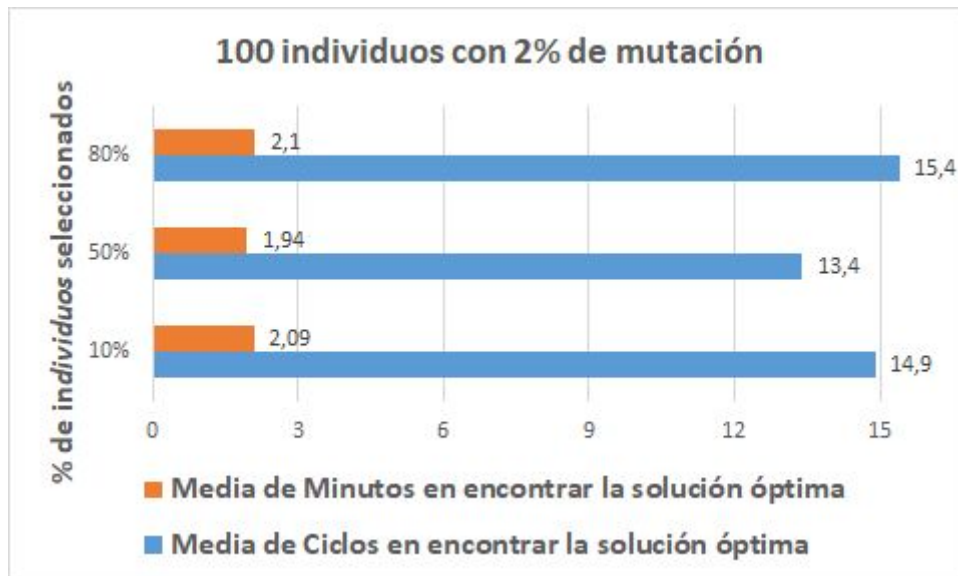
alejarse demasiado de la opción del 3% (la peor hasta el momento anterior de esta última prueba), deduzco que encontrar la solución óptima en el caso en que se tienen 4 estaciones no depende demasiado del porcentaje de individuos que se elijan en los torneos, pero cuanto más cerca se esté de elegir el 50% de los individuos, mejores resultados se obtendrán en cuanto a número de ciclos que se tarda en encontrar la solución óptima se refiere, aunque si se tiene en cuenta el tiempo en encontrar dicha solución, con un 10% se tardan 0,04 minutos menos (2,4 segundos) como se puede apreciar en la siguiente gráfica:



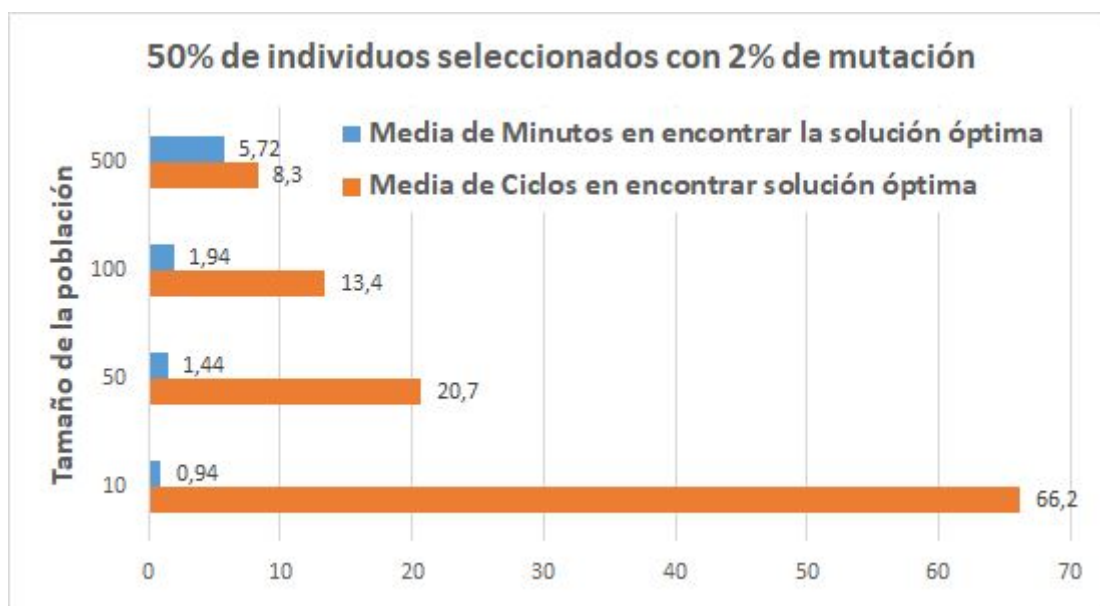
Llegado a este punto, decido cambiar el porcentaje de mutación a 2%, y empiezo probando con un 50% de los individuos seleccionados en los torneos, ya que es el valor de dicho parámetro que me ha dado una solución más satisfactoria en cuanto a los ciclos. Consigo encontrar la mejor solución en $13.4 \approx 13$ ciclos, obteniendo así la solución en menos ciclos que con un 1% de mutación, y además en menos tiempo.



Pruebo con un 10% y un 80% de individuos seleccionados para ver cómo se comporta el algoritmo al tener un 2% de mutación, y obtengo la solución óptima en $14.9 \approx 15$ ciclos con la primera combinación de parámetros y en $15.4 \approx 15$ ciclos con la segunda, por lo que el mejor valor de dicho parámetro sigue siendo 50% si tenemos en cuenta los ciclos, y además cómo se puede ver en la gráfica el tiempo medio también es el menor de todos.



Finalmente decido cambiar el número de individuos, manteniendo los demás parámetros en 50% y 2%. Pruebo además de con 100 individuos, con 10, 50 y 500. Descubro que cuanto menos individuos tiene la población, más ciclos de media tarda en encontrar la solución óptima, y viceversa, por lo que en cuanto a los ciclos la mejor combinación sería con 500 individuos. Calculo los tiempos medios, y ocurre justo al contrario, cuanto menos individuos menos minutos tarda el algoritmo en encontrar la solución. Por tanto, en cuanto al tiempo que se tarda en encontrar la solución óptima, la mejor combinación es con 10 individuos.



En esta tabla muestro los resultados y valores de los parámetros necesarios para obtenerlos:

Tamaño de población	Porcentaje de torneo	Porcentaje de mutación	Media ciclos en encontrar óptima	Media minutos en encontrar óptima
100	3%	1%	19,8	2,4 min
100	5%	1%	16,5	2,7 min
100	10%	1%	14,1	1,96 min
100	50%	1%	13,9	2 min
100	80%	1%	16,8	2,25 min
100	100%	1%	22,4	3,55 min
100	10%	2%	14,9	2,09 min
100	50%	2%	13,4	1,94 min
100	80%	2%	15,4	2,1 min
10	50%	2%	66,2	0,94 min
50	50%	2%	20,7	1,44 min
500	50%	2%	8,3	5,72 min

3.2. Segunda situación (24 estaciones)

Para esta segunda situación he podido hacer menos pruebas teniendo en cuenta que al tener cadenas de 348 bits el tiempo de ejecución aumenta bastante, y además solo he encontrado soluciones sub-óptimas.

Empiezo haciendo la prueba con 1000 individuos, (cabe destacar que esta prueba la hice antes de las de la primera situación) suponiendo que se necesitan más individuos al ser más largos los cromosomas, y con 5% de individuos seleccionados en los torneos y 1% en el porcentaje de mutación. Lo dejo ejecutando durante más de un día (*≈ 32 horas*), y obtengo el mejor fitness en el ciclo 713 siendo este de 0.83366492869 y obteniéndose con el cromosoma:

“00110101001010111100000110111111010100100100001010100101001111000010000101010111100001001011100001000110101101101101101100111100101011010001100111100000100111110000110101000011100001010110101110000001010101011000010010111011100001000100111000100011111001111000101100110101000100010011010100010111000011000010011100010000111110011110001011001101010001000100110101000010111000011000010111100”.

Después de esa prueba intento afinar el fitness fijándome en las pruebas de la situación en que se tienen 4 estaciones, por lo que el porcentaje de individuos seleccionados lo pongo al 50% y el de mutación al 2%. Pruebo con 500 individuos y obtengo el mejor fitness en el ciclo 215 siendo este de 0.837253734769 y obteniéndose con el cromosoma:

“00110101001010101100000110111111010100100110001010100011000111000010000101010011011101010001100000110011010110110110001111011100101011010001100111000001001111111000110101001011100001010101010000001010101011000010001001111110011010000111100000000111100100000101011110110100101111100000000010011100111010111110101110001011001101010001000100010101000101011100010011100”

El fitness encontrado es peor que en la anterior prueba, ya que no me podía permitir esperar tanto, así que bajo a 100 individuos para que el algoritmo vaya más rápido. Encuentro el mejor fitness en el ciclo 564 siendo este de 0.828708913009 y obteniéndose con el cromosoma:

“00110111001010111101000110111111000010010011000101010111011111000010100101110111110001000011100101101011010110001110011100011010001100101001000001111110000110100000011100001010101111000001010111100000101011100001000100111010001101100111110000000010011001000001010111111010010001001000000000000000010011000011111101111100010110011000100011001000101101000101111010011000010111100”

El fitness encontrado en esta prueba es el mejor de las 3, pero para ver si consigo uno mejor, bajo la población a 50 individuos, ya que podré ver más ciclos en menos tiempo. Dejo ejecutando el algoritmo durante 1000 ciclos, pero el algoritmo se estanca y el mejor fitness se encuentra en el ciclo 673 siendo este de 0.837788226546 y obteniéndose con el cromosoma:

“000101110010101111010001101111111010100110110001010101101001111000010100101010111010100010100100000110010110100110110101110111100111000110100011001110000000001111100001101010000111000010101101011100000010101000110000110010011111000110110011110000000000101011110110000100011111010000100010011100100000110111101111000001110110100010001001101000000101111000111000010111100”

El fitness es el peor de las 4 pruebas, por lo que la combinación con la que se consigue el mejor fitness es con 100 individuos, 50% en los torneos y 2% de mutación.

Tamaño de población	Porcentaje de torneo	Porcentaje de mutación	Media ciclos en encontrar sub-óptima	Media horas en encontrar sub-óptima	Solución sub-óptima encontrada
50	5%	2%	673	1,37 h	0.837788226546
100	50%	2%	564	2,32 h	0.828708913009
500	50%	2%	215	4,38 h	0.837253734769
1000	50%	1%	713	31,65 h	0.83366492869

4. Conclusiones y problemas encontrados

Esta práctica ha sido muy enriquecedora, ya que hemos aprendido a hacer un algoritmo genético desde cero, y me ha hecho entender muy bien cómo funciona un algoritmo de este tipo. Además me parece fascinante como el algoritmo encuentra soluciones que con fuerza bruta sería imposible debido a la gran cantidad de combinaciones.

En cuanto a las dos situaciones estudiadas, he notado una gran diferencia en la dificultad de encontrar una solución óptima en la que tiene las 24 estaciones respecto a la que tiene tan solo 4. Esta última he encontrado el fitness *0.0* en todas las pruebas sin problema, a diferencia de la primera, en la que el mejor fitness que he conseguido ha sido *0.828708913009*.