Análisis de Hiperparámetros de un Algoritmo Evolutivo Aplicado al Problema de la Suma del Subconjunto

Andrés Matesanz

Abstract - En este informe se presenta un análisis sobre el rendimiento de un algoritmo evolutivo sencillo sobre el problema *de la suma del subconjunto*. Se presenta el mencionado problema y cómo se ha aproximado su solución. Se realizan variaciones en los valores de los hiperparámetros y su relación con el resultado final. Dado que el número de alelos de cada individuo se ha mantenido invariable durante todo el estudio, se presenta junto a este un caso similar del problema *OneMax* donde se puede apreciar cómo el genotipo afecta al número de iteraciones.

1. Introducción

En matemáticas y ciencias de la computación, un problema de optimización es aquél en el que se trata de encontrar la mejor solución dentro de todas las soluciones factibles a dicho problema.

Los problemas de optimización con múltiples variables discretas (o continuas si se discretiza su espacio) pertenecen a los conocidos como **problemas de optimización combinatoria**. La solución a un problema de optimización es tal que ninguna otra solución dentro del espacio de búsqueda minimiza más la función objetivo. Explorar todo el espacio de búsqueda, en general, es altamente costoso en cuanto a recursos de tiempo y computación, por ello se emplean **algoritmos de aproximación** conocidos como metaheurísticas.

Los **algoritmos evolutivos** son un subconjunto de metaheurísticas, dentro de los problemas de optimización, inspirados en la evolución natural de las especies. Se genera un número determinado de individuos, llamado población, con configuraciones distintas entre sí, cuyo objetivo común es minimizar la función objetivo dentro del espacio de búsqueda. Cada población nace de su predecesora, aplicando una serie de técnicas de cruce y mutación a aquellos individuos que mejores resultados hayan demostrado en cada iteración.

En este caso concreto se va a tratar con el problema *de la **Suma del Subconjunto***. Es uno de los más básicos dentro de la teoría de la complejidad computacional y nos permite mostrar fácilmente cómo funciona un algoritmo evolutivo. Consiste en, dado un conjunto de números ¿existe algún subconjunto cuya suma sea igual a una **cifra dada** *s*? Para resolver este problema se crea un conjunto de individuos con un

genotipo, llamado también cromosoma, formado por uno o varios vectores unidimensionales de alelos, los cuales pueden tomar únicamente los valores 0 y 1. Cada uno de estos alelos es multiplicado uno a uno por cada nçúmero del conjunto mencionado anteriormente (tanto el genotipo como el conjunto deben tener la misma dimensionalidad). La **función objetivo** que se trata de minimizar es la de dar con un individuo que tenga en su **cromosoma** algún vector compuesto únicamente por **valores de 0 y 1 que al multiplicar por el conjunto dé como resultado *s***.

Los alelos, o genes, de los individuos son inicializados aleatoriamente y en cada iteración sufren cruce, con un probabilidad relativamente alta (del 80% para este caso en concreto), y/o mutación con una probabilidad relativamente baja, indirectamente proporcional al número de alelos total que posee cada individuo. La función de *fitness* de cada individuo, por el que se valora su rendimiento en la iteración, se determina en función a la proximidad a *s* derivada de multiplicar sus alelos por el conjunto dado, naturalmente, el *fitness* máximo que puede alcanzar cualquier sujeto de la población es igual a *s*.

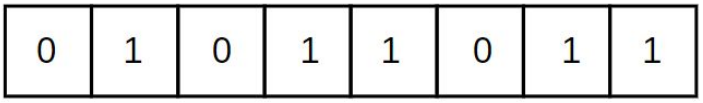


Imagen 1: Ejemplo de un vector de alelos o gen.

2. Estudio empírico

El objetivo de este documento es analizar cómo se comporta el algoritmo evolutivo en función de los hiperparámetro seleccionados previa resolución del ejercicio. Algunos de ellos han sido ya mencionados previamente: longitud del vector de alelos, número de vectores, tamaño de la población, probabilidad de cruce, probabilidad de mutación o número máximo de iteraciones posible. Como se mencionaba anteriormente, el genotipo y el conjunto deben tener igual tamaño: en este caso el conjunto tiene una longitud de 128 cifras, por lo que el resultado de multiplicar el número de genes por su longitud debe ser también de 128.

Los hiperparámetros para los que vamos a estudiar el comportamiento del algoritmo evolutivo son tres: el **número de genes**, la **longitud de cada gen** (interrelacionados de manera que el producto de ambos sea siempre de 128) y el **tamaño de la población.** Son tres valores que a priori están muy interrelacionados, dado que un mayor número de genes amplía el espacio de búsqueda, una población más grande diversifica más las posibilidades de localizar una solución óptima. Queremos investigar **cómo afectan estas variables al número total de iteraciones**.

La simulación del algoritmo evolutivo se puede encontrar aquí. Se trata de un modelo escrito en Java que nos devuelve el número de evaluaciones realizadas en

cada simulación. Una evaluación consiste en determinar el *fitness* de un único individuo. Para este experimento, sin embargo, se ha tomado **el número de iteraciones medio** tras cada simulación para un tamaño poblacional determinado con una longitud de genes determinada, ya que es más representativo del número de pasos de cruce y mutación que ha sufrido la población. Dicho número de iteraciones se extrae dividiendo el total de evaluaciones entre el total de individuos obtenido tras cada ejercicio.

Cabe mencionar, que a pesar de que la variable **tiempo** no ha sido tomada para este ensayo, a mayor número de individuos se experimentó una duración mayor de cada simulación. Es decir, que un número menor de iteraciones no implica una un menor tiempo en la ejecución del algoritmo.

Para este ensayo se ha tomado un conjunto de configuraciones para el **número de genes** y la **longitud de cada gen** aumentando, en el primer caso, y disminuyendo en el segundo, exponencialmente de 1 hasta 128 para que el producto de ambas siempre sea igual a la longitud del conjunto dado. El **tamaño de la población** se ha aumentado exponencialmente desde 2 individuos hasta 512. Cada configuración posible entre estas 3 variables se ha repetido un total de 50 veces hasta obtener significancia estadística.

	Gn: 1 Gl: 128	**Gn: 2 Gl: 64**	**Gn: 4 Gl: 32**	**Gn: 8 Gl: 16**	**Gn: 16 Gl: 8**	**Gn: 32 Gl: 4**	**Gn: 64 Gl: 2**	**Gn: 128 Gl: 1**
Pob. 2	49801,00	96400,25	128338,50	54449,00	52578,00	57822,00	73409,25	78834,75
Pob. 4	36977,63	28687,38	51040,50	21827,75	45905,50	42155,75	45520,00	29858,75
Pob. 8	10677,19	14793,25	14996,13	29830,31	13429,13	12186,56	12300,94	25258,75
Pob. 16	11183,59	8485,75	8561,16	8990,63	8733,53	8447,44	6497,91	6540,81
Pob. 32	4544,59	3184,83	4847,67	5420,05	5484,92	5905,89	3286,03	4309,41
Pob. 64	3891,29	655,93	2716,45	1488,09	2381,52	4191,44	2211,85	2259,61
Pob. 128	1164,81	1105,59	1014,46	1212,30	1335,16	822,98	710,89	865,90
Pob. 256	337,90	212,77	125,43	506,84	183,32	284,22	334,12	563,57
Pob. 512	98,89	73,52	139,58	358,15	78,73	102,06	43,88	243,05

Tabla 1: Mediana de iteraciones según número de individuos y genes

Cada configuración posible ha sido ejecutada en un total de 50 ocasiones para conseguir relevancia estadística de los datos.

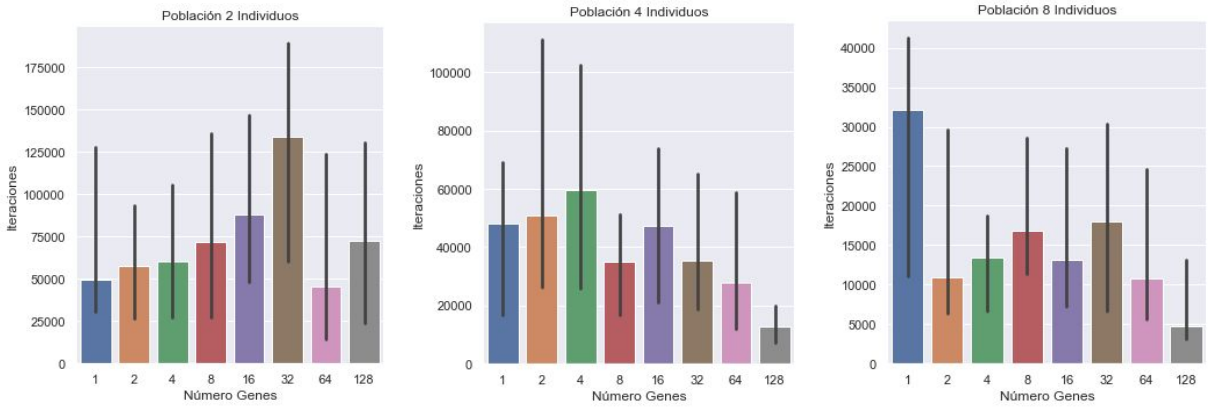
La *tabla 1* muestra la distribución de la **mediana del número de iteraciones** dado un tamaño poblacional con una longitud y un número de genes determinado, mientras que la *tabla 2* muestra la desviación estándar de las mediciones.

	Gn: 1 Gl: 128	Gn: 1 Gl: 128	Gn: 1 Gl: 128	Gn: 1 Gl: 128	Gn: 1 Gl: 128	Gn: 1 Gl: 128	Gn: 1 Gl: 128	Gn: 1 Gl: 128
Pob. 2	295544,93	146717,09	190723,24	246507,15	138903,61	204983,79	231010,38	213707,96
Pob. 4	127078,06	128552,08	143884,35	70557,56	142462,19	93285,50	93126,19	115090,10
Pob. 8	31625,40	36014,80	45120,60	65825,46	45562,12	41513,86	64028,84	54176,70
Pob. 16	32570,80	22066,16	15366,96	21815,46	15591,88	31406,27	28053,63	39479,71
Pob. 32	15216,27	14589,87	8133,15	8940,25	16145,44	9794,36	14651,22	20528,24
Pob. 64	5959,34	4841,80	7273,36	5062,89	4239,93	6278,65	6304,74	5850,35
Pob. 128	2217,41	3393,95	2416,37	3337,64	2803,99	4411,43	3418,48	2866,04
Pob. 256	1425,66	1354,95	1407,33	1305,23	1308,35	1109,72	2202,40	1572,77
Pob. 512	444,67	813,14	805,21	963,87	1004,61	557,88	646,99	559,93

Tabla 2: Desviación estándar de iteraciones según número de individuos y genes

En la *imagen 2* se puede apreciar mejor la tabla anterior. La variable *y* muestra la **mediana de las iteraciones** y su desviación típica para cada uno de los casos, mientras que la variable *x* está determinada por **el número de genes** total que posee cada individuo de la población.

Se debe apreciar que tanto una variable, como la otra, están dispuestas en forma **logarítmica**. Es preciso recordar también que el número de genes y su longitud se encuentran inversamente relacionados para que su producto sea constante, por ello en el presente estudio no se muestra gráficamente la relación entre la longitud de los genes y el número de iteraciones.



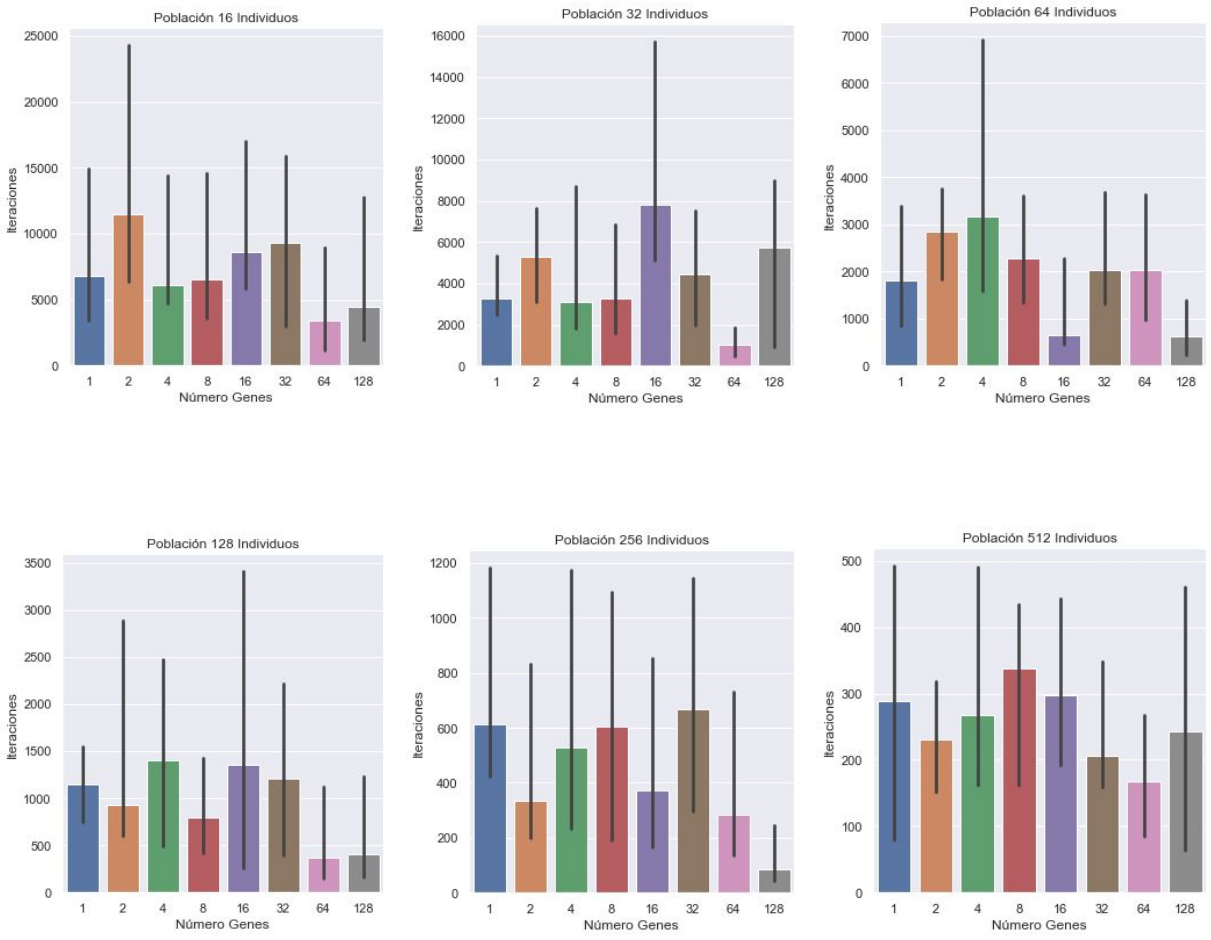


Imagen 2: Distribución de iteraciones según tamaño poblacional y número de genes

A continuación vamos a observar gráficamente cómo influyen las distintas variables en el número total de iteraciones **independientemente** entre sí.

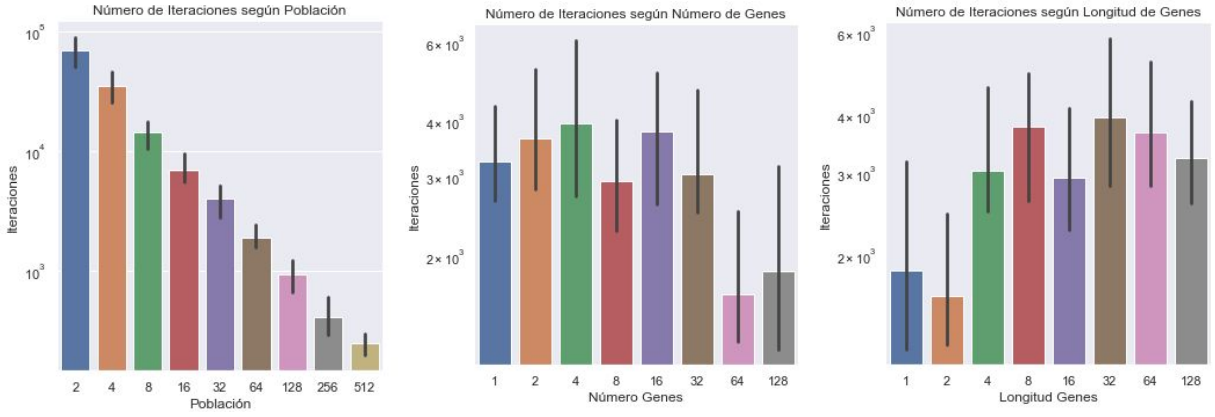


Imagen 3: Iteraciones según las distintas variables analizadas

3. Resultados

De estas gráficas se pueden extraer una serie de conclusiones. En primer lugar podemos concluir que mientras el **producto** entre el **número de genes y su longitud** para cada individuo **sea constante no existe correlación con el número de iteraciones resultante**.

En segundo lugar, como se mencionaba anteriormente, **una población de tamaño menor implica una mayor exploración del espacio de búsqueda**, por ello, en la configuración poblacional con 2 individuos se obtiene el número más alto de iteraciones registradas.

En tercer lugar, viendo las distintas gráficas, tanto el eje x como el eje y están dispuestos en escala logarítmica, por lo que a medida que aumenta exponencialmente el número de genes lo hace también, pero de forma inversa, el número de iteraciones. Podemos concluir que **hay una relación lineal inversa entre el número de iteraciones y el tamaño de la población.**

En cuarto lugar, la dispersión del número de iteraciones registradas según la configuración del genotipo es muy amplia

Por último tenemos la siguiente imagen, que acumula en una sóla todas las gráficas presentadas en la imagen 2, donde se puede ver con mayor claridad los resultados encontrados en este ensayo:

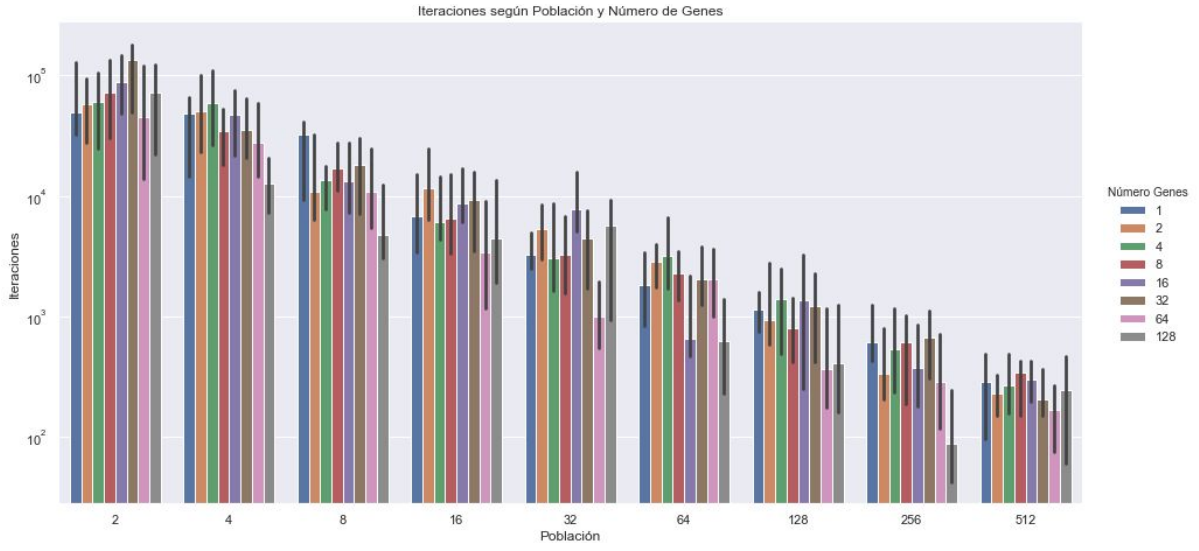


Imagen 4: Comparativa de los Resultados

4. Conclusiones

Para este caso concreto del problema de la *Suma del Subconjunto* se sabe que existe una solución y es conocida. Sin embargo los algoritmos evolutivos pertenecen al conjunto de algoritmos de aproximación dentro de los problemas de optimización combinatoria, para los cuales no siempre se puede determinar que la solución encontrada es la que más minimiza la función objetivo.

Dado que el número de alelos de cada individuo se ha mantenido invariable durante todo el estudio, se presenta junto a este un caso similar del problema *OneMax* donde se puede apreciar cómo el genotipo afecta al número de iteraciones.

Con este tipo de ejemplos podemos visualizar cómo se comportan dichos algoritmos dentro de una casuística controlada, lo que nos ayuda a aplicarlos a problemas mucho más complejos.

5. Referencias

Los resultados obtenidos y presentados en este informe pueden verse recogidos en el siguiente repositorio.