

**OPTIMIZACIÓN HEURÍSTICA
EVALUACIÓN DE ALGORITMOS
EVOLUTIVOS**

Grupo 08

Ulises Iago Bértolo García

José López López

Mihail Neagu

Algoritmos seleccionados para la realización de la práctica

En esta práctica hemos implementado dos algoritmos. Para el primero de ellos hemos escogido un algoritmo evolutivo diferencial ya que en la anterior habíamos desarrollado un algoritmo evolutivo genético. Los parámetros de nuestro nuevo algoritmo se dividen en varias categorías.

El primer parámetro que se pondrá en ejecución será el operador de selección. Se encarga de seleccionar los vectores ‘target’ y ‘donors’ sin colisiones. Los vectores ‘target’ se irán escogiendo en orden en cada iteración y los ‘donors’ a través de posiciones aleatorias de la población.

El segundo parámetro es el operador de mutación, que en nuestro algoritmo es el operador de rand/1. Utiliza los tres vectores ‘donor’ y les aplica una función para mutación para posteriormente devolver un genoma mutado a partir de los tres anteriores.

El tercer parámetro es el operador de cruce, que será BinomialCrossover, y generará un vector a partir de los que se le pasarán como argumentos. Para ello utilizará una distribución $U[0,1]$ y una constante ‘CR’ que seleccionarán cada elemento de los dos vectores para añadir al resultado.

El último parámetro en ejecución es el operador de reemplazo. En nuestro caso seguirá una política de reemplazo elitista, eligiendo siempre al mejor individuo entre las dos poblaciones actual y nueva.

El segundo algoritmo seleccionado será un algoritmo evolutivo genético de la biblioteca ‘deap’ que tendrá también divididos sus parámetros en varios tipos.

El operador de selección será el de torneo, el de mutación el gaussiano y el de cruce el uniforme.

Resultados obtenidos

Para obtener los datos de los dos algoritmos que hemos escogido hemos ejecutado cada uno con los parámetros siguientes. El primero son las funciones, que son once y sus nombres son; “sphere”, “ackley”, “rosenbrock”, “rastrigin”, “griewank”, “schwefel_2_21”, “schwefel_2_22”, “schwefel_1_2”, “extended_f_10”, “bohachevsky” y “schaffer”. EL tamaño de la población de los algoritmos será de 50, el área de búsqueda es un intervalo de $(-10, 10) * 5$ y el número de iteraciones por función es de 2500.

La primera ejecución será la de nuestro algoritmo evolutivo DE. Los resultados estadísticos obtenidos se muestran en la tabla a continuación.

| DE IMPLEMENTADO | Media | Desv. Tipica | Mínimo | Máximo | Mediana |
|-----------------|-------------|--------------|----------|----------|----------|
| ackley | 2.26E-01 | 0.524791213 | 6.57E-12 | 1.89E+00 | 2.53E-03 |
| bohachevsky | 0.123093418 | 0.175150711 | 1.46E-27 | 4.18E-01 | 5.65E-03 |
| extended_f_10 | 3.194569139 | 2.550989701 | 2.00E-01 | 7.97E+00 | 2.24E+00 |
| griewank | 1.000038528 | 0.00011209 | 1.00E+00 | 1.00E+00 | 1.00E+00 |
| rastrigin | 11.43880997 | 5.825935714 | 3.72E+00 | 2.65E+01 | 1.05E+01 |
| rosenbrock | 31.78204413 | 33.02139321 | 2.03E-15 | 1.22E+02 | 2.51E+01 |
| schaffer | 2.489783066 | 1.512595619 | 4.74E-01 | 6.12E+00 | 2.54E+00 |
| schwefel_1_2 | 3.776191263 | 9.723095567 | 5.28E-73 | 3.55E+01 | 1.64E-02 |
| schwefel_2_21 | 0.444425154 | 0.901053317 | 3.40E-21 | 3.08E+00 | 1.43E-03 |
| schwefel_2_22 | 0.077805822 | 0.176712215 | 2.40E-12 | 6.46E-01 | 7.77E-05 |
| sphere | 0.589590622 | 1.265421546 | 3.12E-36 | 4.50E+00 | 8.27E-06 |

Como se puede observar, los resultados suelen tender al valor cero (valor óptimo), como es el caso de la función “ackley” en la que ha tenido el mejor desempeño medio. También hay algunas funciones como “rosenbrock” en las cuales probablemente nuestro algoritmo no sea suficientemente avanzado, ya que, aunque ha conseguido un mínimo muy cercano a cero, la desviación típica es muy elevada al igual que su media que dista mucho del valor esperado.

En general los valores mínimos en todas las funciones son muy aceptables, por lo que en caso de ejecutar con más iteraciones se obtendrían mejores resultados en todas las funciones.

La segunda ejecución será la del algoritmo evolutivo GA de biblioteca. Los resultados estadísticos obtenidos se muestran en la tabla a continuación.

| GA BIBLIOTECA | Media | Desv. Típica | Mínimo | Máximo | Mediana |
|---------------|-------------|--------------|----------|----------|----------|
| ackley | 0.001355544 | 0.001395336 | 5.02E-04 | 5.59E-03 | 7.29E-04 |
| bohachevsky | 1.78976E-05 | 4.00224E-05 | 4.96E-07 | 1.37E-04 | 8.22E-07 |
| extended_f_10 | 0.123516243 | 0.072162464 | 8.03E-02 | 3.15E-01 | 8.82E-02 |
| griewank | 1 | 1.97881E-10 | 1.00E+00 | 1.00E+00 | 1.00E+00 |
| rastrigin | 0.000202074 | 0.00060708 | 7.45E-06 | 2.21E-03 | 7.45E-06 |
| rosenbrock | 1.637536771 | 0.004356151 | 1.63E+00 | 1.65E+00 | 1.64E+00 |
| schaffer | 0.142907382 | 0.023189959 | 1.28E-01 | 2.02E-01 | 1.31E-01 |
| schwefel_1_2 | 0.000145038 | 0.000264967 | 1.75E-05 | 9.73E-04 | 3.40E-05 |
| schwefel_2_21 | 0.000724761 | 0.000407013 | 3.76E-04 | 1.60E-03 | 5.86E-04 |
| schwefel_2_22 | 0.000867412 | 0.000675778 | 2.14E-04 | 2.67E-03 | 7.42E-04 |
| sphere | 8.48115E-07 | 2.42023E-06 | 5.46E-08 | 8.85E-06 | 5.46E-08 |

En la tabla de datos se puede comprobar que los valores son mucho más exactos y se acercan mejor al valor óptimo de las soluciones de cada función. En el caso de “griewank” ha realizado una aproximación casi perfecta, tendiendo al valor 1 con una desviación típica ínfima. Las desviaciones de este algoritmo son mucho menores que las del anterior, lo que es una clara señal de que su desempeño ha sido más eficiente en las mismas iteraciones. La función con peor desempeño ha sido “rosenbrock”, al igual que la del nuestro.

A continuación, realizamos dos test estadísticos para comparar los dos algoritmos. El primero es el test de Kruskal-Wallis, que miden las diferencias de rendimiento de los dos algoritmos.

| Test de Kruskal-Wallis | Valor P |
|------------------------|-------------|
| ackley | 9.54E-01 |
| bohachevsky | 0.561654133 |
| extended_f_10 | 5.10E-05 |
| griewank | 0.486425684 |
| rastrigin | 1.96E-05 |
| rosenbrock | 0.00557329 |
| schaffer | 2.98E-05 |
| schwefel_1_2 | 1 |
| schwefel_2_21 | 1 |
| schwefel_2_22 | 0.355297106 |
| sphere | 0.558526508 |

Los valores de este test tienen una interpretación breve. Si el valor P es menor que 0.05 significa que el rendimiento entre los dos algoritmos tiene varias diferencias muy significativas. Es el caso de “extended_f_10”, “rastrigin”, “rosenbrock” y “schaffer”, siendo el segundo el menor de todos. Estos cambios en el rendimiento de los dos algoritmos son debidos a sus diferencias para encontrar las soluciones óptimas.

Por último, hemos realizado el test de Wilcoxon, que nos muestra un único P valor que compara el rango medio de las dos muestras relacionadas y determina sus diferencias.

| | |
|-------------------------|-------------|
| Test de Wilcoxon | Valor P |
| | 0.000976563 |

Este P valor es menor que 0.05, lo que significa que la diferencia entre las medianas de las poblaciones y la hipotética es significativamente diferente.

Conclusiones obtenidas

De los datos comentados se pueden sacar varias conclusiones en claro. La primera es que nuestro algoritmo no es el óptimo ni en el tiempo que requiere ni en los resultados que devuelve. El algoritmo necesitó varias horas de ejecución continua para conseguir los valores con los parámetros antes mencionados, y no consiguió los valores óptimos esperados. En cambio, el GA de biblioteca consiguió unos buenos resultados en mucho menor tiempo.

Bajo nuestro punto de vista, esta lentitud e inexactitud se pueden deber a varios factores. Algunos como una mejor optimización del código o también conocimientos más avanzados en la programación de algoritmos evolutivos habrían ayudado a mejorar estos parámetros.

Aún así, creemos que los resultados han sido prometedores, y que con más iteraciones nuestro algoritmo habría conseguido unos resultados óptimos para todas las funciones, aunque se necesitaría una máquina potente para reducir el tiempo necesario.