

Proyecto Metaheurísticas

Annie Hernández Pérez
C – 411

Como proyecto final del curso optativo de Metaheurísticas se implementó en *Python* la metaheurística *Evolución Diferencial*, con el objetivo de mejorar los resultados obtenidos por *Leaders and Followers* en la optimización de la función de *Rastrigin*. Específicamente se decidió implementar *DE/rand/1/exp*, comenzando con una población inicial distribuida de manera uniforme y aleatoria. En esta se obtiene, por cada individuo de la población, un vector de prueba mediante la mutación que luego será usado, junto con el individuo de la población que estamos analizando, por el operador de recombinación para obtener un descendiente.

Como operador de mutación se usa:

$$trial = x_i + f * (x_j - x_k)$$

Donde x_i, x_j, x_k son tres individuos de la población, elegidos aleatoriamente, que son diferentes entre ellos y diferentes respecto al individuo actual que estamos analizando; y f es un parámetro que determina el factor de amplificación del vector diferencia. Este enfoque para el operador de mutación dió mejores resultados para el problema actual que otros como *best*, donde el vector diferencia se adiciona al mejor individuo de la población, u otros donde se usan una mayor cantidad de vectores diferencias.

Se usa un operador de recombinación binomial, este se aplica entre el individuo que se analiza actualmente y el individuo de prueba obtenido para obtener un descendiente. Consiste en que, para cada elemento del descendiente, vamos a decidir con una determinada probabilidad si vamos a poner el elemento del individuo que estamos analizando o el del de prueba; para evitar que el descendiente sea igual al individuo que ya estamos analizando se fuerza a que tenga al menos una componente igual al individuo de prueba. Para lograrlo se genera un número aleatorio de manera uniforme entre 0 y 1; si este es menor que un parámetro cr , la probabilidad de recombinación, se pone la componente del individuo creado durante la mutación en el descendiente, de lo contrario se pone la del individuo que estamos analizando.

Finalmente se evalúa el descendiente obtenido, el que tenga mejor evaluación entre el padre y el descendiente es el que pasa a la próxima generación. Una vez que terminamos de analizar todos los individuos de la población es necesario ajustar los parámetros de control f y cr , sus valores iniciales son 0,8 y 0,5 respectivamente, para la próxima iteración y así obtener mejores resultados. Este ajuste se realiza de la manera siguiente:

```

if  $U \sim (0, 1) < 0,1 :$ 
 $f = 0,1 + U \sim (0, 1) * 0,9$ 
endif

```

```

if  $U \sim (0, 1) < 0,1 :$ 
 $cr = U \sim (0, 1)$ 
endif

```

De manera que se modifican los valores de f entre $[0,1,1]$ y los de cr entre $[0,1]$ de manera aleatoria con una probabilidad de $1/10$. El otro parámetro utilizado por el algoritmo es el tamaño de la población y fue establecido como $2 * dim + (30 - dim)$, siendo entonces el doble de la dimensión del problema en el caso de 30 dimensiones, debido a que fue el valor que brindaba mejores resultados. Con esta implementación especificada anteriormente se tuvieron mejores resultados que los de *Leaders and Followers* para la función propuesta, obteniéndose como mínimo 10 cifras significativas correctas en todas las ejecuciones realizadas en 30 dimensiones.