PRÁCTICA HITO 2: ALGORITMOS MULTI-OBJETIVO

Carlos Camacho Gómez

Francisco Serradilla García

David Camacho Fernández

1.Normativa

Esta práctica evaluable está sujeta a una normativa enumerada en los siguientes puntos:

- La práctica se realizará en grupos, los mismo que para el hito 1.
- La práctica consiste en la realización de los hitos 1(ya realizada) y 2 (detallada en este documento).
- La realización de la práctica se hará en horario de clase.
- La evaluación del hito 2 se hará en la semana 16 (9 de enero de 2025) de 18:30 a 21:00 horas, en el aula.
- Cada grupo deberá realizar una presentación (15 minutos) en donde: se explique el algoritmo implementado; se recojan los resultados obtenidos; se justifiquen las decisiones tomadas a la hora de diseñar el algoritmo evolutivo; se justifiquen los cambios, si los hubiese, resultantes de aplicar dicho algoritmo a una función de fitness u otra; y se detalle el reparto de trabajo realizado dentro del grupo.
- Los profesores se reservan el derecho a realizar preguntas a los integrantes del grupo durante y al final de la exposición.
- Se valorará la calidad de los resultados obtenidos.
- Se valorará la calidad de la presentación, su estructura, organización, análisis de los resultados y las conclusiones presentadas.
- Tanto el código desarrollado como la presentación deberá subirla **un único miembro por cada grupo** a la tarea de Moodle que encontraréis en el apartado del tema 3.
- La fecha de cierre de esta tarea será el jueves 9 de enero a las 18:00.
- La puntuación de este hito tendrá un máximo de 10 puntos y un mínimo de 0.
- Para aprobar la asignatura es necesario obtener un 5/10 pts en la nota de este hito.

2.Enunciado

Después de completar el primer hito, habéis aprendido a diseñar algoritmos meta-heurísticos y los habéis utilizado en varios problemas objetivo. Esos problemas se caracterizaban por tener una única solución óptima, y vuestro objetivo y el de vuestros algoritmos, era encontrarla.

En este hito vais a adaptar y utilizar vuestros propios algoritmos (los del hito anterior) en resolver problemas de naturaleza multi-objetivo. En este tipo de problemas, existen un conjunto de soluciones que no se dominan entre sí, es decir, que son óptimas. A este conjunto se le llama **Pareto-optimal**. Los valores de las funciones de fitness representado en su espacio forman lo que se denomina **Frente de Pareto**. Vuestro nuevo objetivo y el de vuestros algoritmos, es encontrar ese conjunto de soluciones óptimas.

En este hito vais a resolver un total de cinco problemas de optimización y os vais a comparar con el algoritmo NSGA2, el rival a batir. Usaremos la librería **Pymoo** ya que nos ofrece una implementación de 4 de los 5 problemas de optimización y del algoritmo NSGA2. La interfaz de usuario de Pymoo está diseñada para ser bastante sencilla e intuitiva, especialmente para aquellos que ya tienen algo de experiencia con Python y la optimización. Como podréis ver a través de este enlace, cuenta con una documentación detallada y ejemplos claros que ayudan a los usuarios a entender cómo utilizar las diversas funcionalidades de la librería. También nos permite crear nuestros propios problemas personalizados e integrarlos de manera fácil. Por último, nos ofrece una herramienta para visualizar nuestros resultados e interpretarlos.

A continuación, se detallan una serie de indicaciones que puedes seguir para la elaboración del hito:

- Lo primero que debéis hacer es adaptar uno o varios de los algoritmos del hito 1 para problemas multi-objetivo. Los cambios irán desde modificar el código para que trabaje con no una sino dos valores de fitness por cada individuo, hasta implementar la heurística de selección de la siguiente población (en principio una de las vistas en clase, pero podéis ser creativos).
- Por supuesto podéis introducir si lo deseáis nuevas mejoras en las formas de cruce y/o mutación, se valorará positivamente.
- La implementación se llevará a cabo en un notebook de Jupyter y se ejecutará sobre Python 3. Podéis descargar un cuadernillo desde el Moodle de la asignatura, que contiene una base para el uso de la biblioteca Pymoo.
- Como sabéis, estos algoritmos son estocásticos. Por tanto, debéis lanzar al menos 10 experimentos y sacar algunos estadísticos (a elegir por el grupo) para mostrar los resultados en las presentaciones.
- Los resultados contarán de al menos una tabla donde se recojan al menos dos métricas de las vistas en teoría para comparar y evaluar la calidad de los Frentes de Pareto.
- Los resultados deben ir acompañados de una gráfica donde se muestren los paretos del algoritmo NSGA2 y del mejor experimento (o uno de los mejores) de vuestro algoritmo.
- Es opcional introducir nuevos algoritmos en la comparativa, pero se valorará positivamente.
- Los presupuestos (*budget*) de los algoritmos están asociados a cada problema objetivo, que se detallan a continuación.

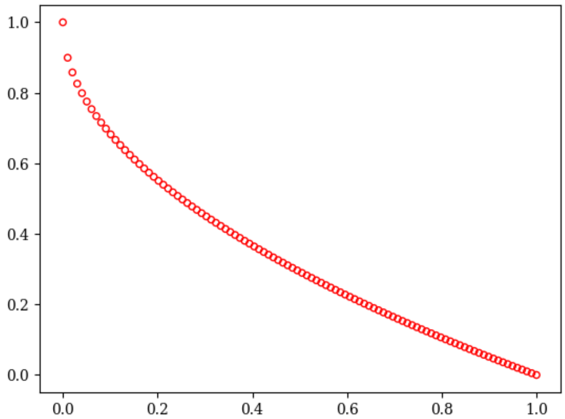
2.1-Problemas de optimización

Las funciones a optimizar son:

1. ZDT1: Es una función con treinta variables cuyo frente de Pareto es convexo. La función se expresa matemáticamente como:

$$
\begin{aligned}
f_1(x) &= x_1 \\
g(x) &= 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i \\
h(f_1, g) &= 1 - \sqrt{f_1/g}
\end{aligned}
$$

$$0 \leq x_i \leq 1 \quad i = 1, \dots, n$$

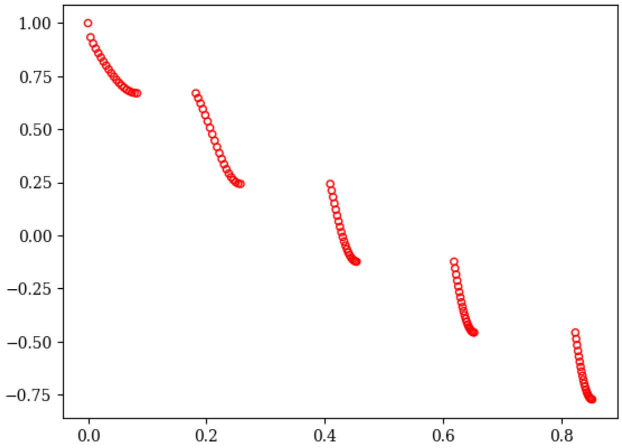


El espacio de búsqueda de las variables será [0,1].
El número máximo de evaluaciones de la función de fitness será 10k.

2. ZDT3: Es una función con treinta variables con un frente de Pareto separable. La función se expresa matemáticamente como:

$$
\begin{aligned}
f_1(x) &= x_1 \\
g(x) &= 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i \\
h(f_1, g) &= 1 - \sqrt{f_1/g} - (f_1/g)\ \sin(10\pi f_1)
\end{aligned}
$$

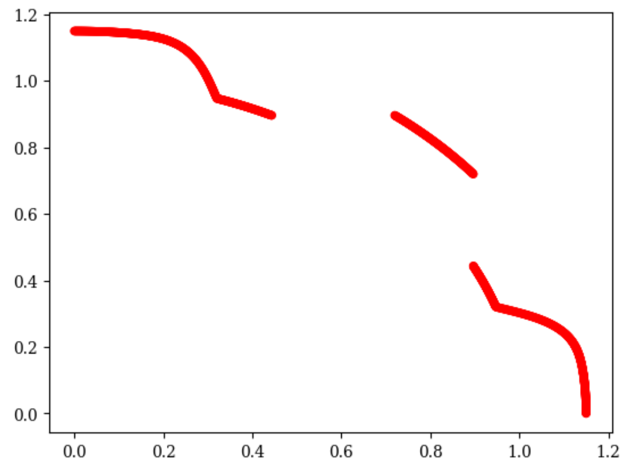
$$0 \leq x_i \leq 1 \quad i = 1, \dots, n$$



El espacio de búsqueda de las variables será [0,1].
El número máximo de evaluaciones de la función de fitness será 10k.

3. MW7: Es una función con treinta variables y dos restricciones (c_1 y c_2) que dan lugar a un frente de Pareto separable. La función se expresa matemáticamente como:

$$
\begin{aligned}
&\min \begin{cases} f_1(\vec{x}) = g_3 x_1 \\ f_2(\vec{x}) = g_3 \sqrt{1 - (f_1/g_3)^2} \end{cases} \\
&\text{s.t.} c_1(\vec{x}) = \left(1.2 + 0.4 \sin(4l)^{16}\right)^2 - f_1^2 - f_2^2 \geq 0 \\
&\quad c_2(\vec{x}) = \left(1.15 - 0.2 \sin(4l)^{8}\right)^2 - f_1^2 - f_2^2 \leq 0 \\
&\quad l = \arctan(f_2/f_1).
\end{aligned}
$$



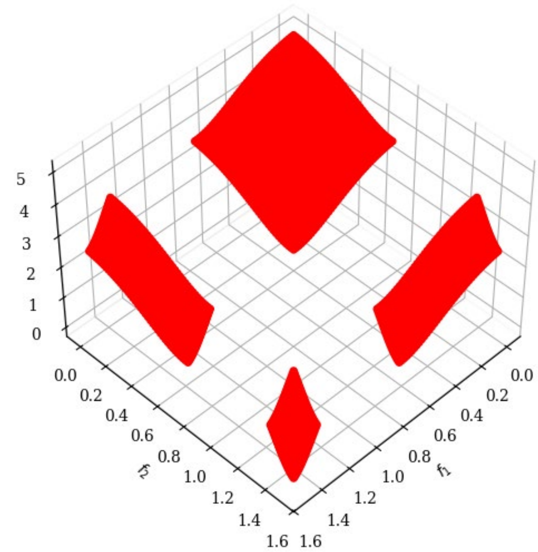
El espacio de búsqueda de las variables será [0,1].
El número máximo de evaluaciones de la función de fitness será 10k.

4. MW14: Es una función para un número de funciones configurable. Nosotros la usaremos con 3 funciones e individuos de treinta variables. Da lugar a un frente de Pareto separable. La función se expresa matemáticamente como:

$$\min \begin{cases} f_{k=1:m-1}(\vec{x}) = x_k \\ f_m(\vec{x}) = g_3/\left(m-1\right)\sum_{i=1}^{m-1}\left(6 - \exp(f_i) - 1.5\sin\left(1.1\pi f_i^2\right)\right) \end{cases}$$

$$\text{s.t.}\, c(\vec{x}) = 1/(m-1)\sum_{i=1}^{m-1}(6.1 - \alpha(i)) - f_m \geq 0$$

$$\alpha(i) = 1 + f_i + 0.5 f_i^2 + 1.5\sin\left(1.1\pi f_i^2\right).$$



El espacio de búsqueda de las variables será [0,1.5].
El número máximo de evaluaciones de la función de fitness será 10k.

5. *Travelling Salesman Problem MO* de N=100 ciudades. El problema trata de minimizar la distancia y el tiempo que tarda un vendedor que tiene que pasar únicamente una vez por cada ciudad y volver a la ciudad inicial.

$$Min\ F(X) = D(x_N, x_1) + \sum_{i=1}^{N-1} D(x_i, x_{i+1})$$

$$Min\ G(X) = T(x_N, x_1) + \sum_{i=1}^{N-1} T(x_i, x_{i+1})$$

Para este problema se partirá de una versión adaptada a Pymoo, donde se trabaja con las matrices de tiempo y distancia (simétrica).
El número máximo de evaluaciones de la función de fitness será 100k.