

1. (30%) Considere el algoritmo descrito en el artículo asignado: "A bi-population EDA for solving the no-idle permutation flow-shop scheduling problem with the total tardiness criterion" (enviado adjunto con el examen). Clasifique sus diferentes componentes de acuerdo a su categoría (diversificación o intensificación).
2. (10%) Indique a cuál metaheurístico corresponde el siguiente pseudo-código.

```

procedure
     $s \leftarrow \text{initial\_solution}()$ 
     $T = T_0$ ;
    while  $T > T_F$  do
         $l = 0$ ;
        while  $l < L$  do
             $l = l + 1$ ;
            Find  $s' \in N(s)$ ;
             $d = f(s') - f(s)$ ;
            if  $d < 0$  do
                 $s \leftarrow s'$ ;
            else
                if  $\text{random} < e^{-d/T}$  do
                     $s \leftarrow s'$ ;
                end;
            end;
        end;
         $T = rT$ ;
    end;
    return  $s$ ;
end
    
```

3. (30%) De acuerdo con el método del numeral anterior, indique cual sería la modificación necesaria si el método aplicara su criterio de aceptación sobre la mejor entre m soluciones generadas.
4. (30%) Dado el siguiente problema de optimización:

$$\begin{aligned}
 \text{Min } Z &= e^{x_1 x_2 x_3 x_4 x_5} \\
 x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 &= 0 \\
 x_2 x_3 - 5 x_4 x_5 &= 0 \\
 x_1^3 + x_2^3 + 1 &= 0 \\
 l_i &\leq x_i \leq u_i \\
 l_i &= -u_i \\
 u_i &= [2.3, 2.3, 3.2, 3.2, 3.2]
 \end{aligned}$$

Y la siguiente relajación del problema:

$$\text{Min } Z^R = e^{x_1 x_2 x_3 x_4 x_5} + e^{g_1(X) + g_2(X) + g_3(X) + h_1(X) + h_2(X) + h_3(X) + h_4(X) + h_5(X)}$$

$$g_1(X) = |x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10|$$

$$g_2(X) = |x_2 x_3 - 5 x_4 x_5|$$

$$g_3(X) = |x_1^3 + x_2^3 + 1|$$

$$h_i(X) = \begin{cases} -u_i - x_i & \text{si } x_i \leq -u_i \\ 0 & -u_i \leq x_i \leq u_i \\ x_i - u_i & \text{si } x_i \geq u_i \end{cases}$$

Considere, además, las siguientes tres soluciones al problema relajado:

	x_1	x_2	x_3	x_4	x_5	Z^R
S1 =	-1,48	-0,07	-2,36	1,13	1,24	15.085,82
S2 =	-0,49	-1,78	2,01	0,58	0,38	103.060,69
S3 =	0,27	2,20	-0,32	0,33	2,47	50.802.969,28

Implemente un algoritmo (en Python o Matlab) que realice combinaciones de las soluciones utilizando el operador de cruce uniforme de Algoritmos Genéticos.