

Ejercicio 4: Algoritmos Genéticos

Se quiere decidir el orden en el que han de consumirse los alimentos de un almacén que están a punto de caducar. Para ello, considere lo siguiente:

1. Hay que seleccionar un alimento a consumir cada día. Todos los alimentos de la lista *AL* han de consumirse.
2. A cada alimento le quedan unos días para caducar (*dC*) por lo que sólo pueden ser consumidos antes de su fecha de expiración.
3. Además, no puede consumirse dos días consecutivos alimentos del mismo *tipo*.
4. Se persigue maximizar la suma de los cuadrados de los días que le quedan para caducar a cada producto al ser consumidos.

Ejemplo:

Para los siguientes datos:

Nombre	Días para Caducar	Tipo
Hamburguesa	3	1
Fajitas	7	1
Macedonia	6	2
Dorada	3	3
Gambas	2	3

Una solución sería:

Orden:

[Fajitas, Gambas, Hamburguesa, Dorada, Macedonia]

El producto del tiempo antes de caducar sería 55, ya que a las Fajitas le quedarían 7 días (se consumen en el día 0), a las Gambas 1 (2 -1), a la Hamburguesa 1 (3-2), a la Dorada 0 (3-3) y a la Macedonia 2 (6-4). Entonces $7^2 + 1^2 + 1^2 + 0^2 + 2^2 = 55$.

Tenga en cuenta que el primer día es el 0.

Ejemplo sin solución:

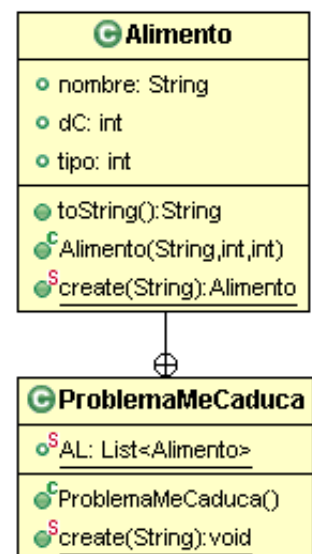
Para los siguientes datos:

Nombre	Días para Caducar	Tipo
Hamburguesa	1	1
Fajitas	1	1
Macedonia	1	2

No tiene solución ya que es imposible colocar tres alimentos a los que sólo le queda un día para caducar. Cómo mucho podrían colocarse dos.

Se pide:

1. ¿Qué tipo de cromosomas son los más adecuados para resolver el problema y por qué?
2. Indique, **razonadamente**, cómo sería la lista de enteros decodificada para el ejemplo del enunciado.
3. Explique y defina, lo más formal que pueda, cuál sería la función de fitness.
4. Implemente los siguientes métodos:
 - a. **public** Integer getObjectNumber()
 - b. **public** Integer getMax(int index)
 - c. **public** Double fitnessFunction(IndexChromosome cr)



1. IndexPermutation ya que todos los elementos tienen que aparecer en la solución cambiando su orden.

[2,0,4,3,1] ya que se coloca en orden cada índice de la tabla AL

$$\begin{aligned}
 \text{productoDias} &= \sum_{i=0}^{i < \text{decode}-1} (AL_{\text{decode}(i)} \cdot dC - i)^2 \\
 \text{caducados} &= \sum_{i=0}^{i < \text{decode}-1} AL_{\text{decode}(i)} \cdot dC < i ? 1 : 0 \\
 \text{tiposConsecutivos} &= \sum_{i=0}^{i < \text{decode}-1} AL_{\text{decode}(i)} \cdot \text{tipo} = AL_{\text{decode}(i+1)} \cdot \text{tipo} ? 1 : 0 \\
 \text{fitness} &= \text{productoDias} - \text{CTE} * (\text{caducados} + \text{tiposConsecutivos})^2
 \end{aligned}$$

Donde CTE es un número suficientemente grande.

2.

```

a) public Integer getObjectsNumber() {
    return AL.size();
}

b) public Integer getMax(int index) {
    return 1;
}

c) public Double fitnessFunction(IndexChromosome cr) {
    List<Integer> ls= cr.decode();

    int sumaDiasAntesCaducar= IntStream.range(0, ls.size())
        .map(i-> (int) Math.pow(AL.get(ls.get(i)).dC-i,2))
        .sum();

    int caducados=(int)IntStream.range(0, ls.size())
        .filter(i-> AL.get(ls.get(i)).dC < i)
        .count();

    int tiposConsecutivosIguales=(int)IntStream.range(0,ls.size()-1)
        .filter(i-> AL.get(ls.get(i)).tipo == AL.get(ls.get(i+1)).tipo )
        .count();

    return (double)sumaDiasAntesCaducar - (100000000* (caducados +
tiposConsecutivosIguales));
}

```