



Algoritmos Genéticos (Opt4J)

Nombre: Adrian Tendero Lara

Importante: subid a Poliformat el código generado para cada pregunta (por ejemplo en un .zip).

1. (5 puntos) Tras múltiples experimentos sobre los materiales resultantes, se ha descubierto que existen combinaciones de tres capas que afectan sustancialmente a las propiedades del material:

- Capas consecutivas (de capa exterior a interior) ABC o ABD aumentan la capacidad de aislamiento total del material en un 2%.
- Capas consecutivas (de capa exterior a interior) FHI o FHJ pueden resultar inflamables. Por tanto, hay que evitar estas combinaciones a toda costa.

Explica el cambio realizado e indica cuál es la mejor solución (combinación de materiales y aislamiento total) que encuentras para este nuevo problema tras la ejecución de 1000 iteraciones. El resto de los parámetros se deja a libertad del alumno (**NOTA:** indicad claramente cuáles son estos parámetros).

NOTA DE IMPLEMENTACIÓN: La clase String tiene un método `indexOf` que nos indica si una subcadena está dentro de otra cadena.

En la función de `CapasEvaluator` he modificado la función resultado si se detecta cobinaciones de genotipos (A,B,C)(A,B,D) (F,H,i) y (F,H,J) alteren el resultado total de manera apropiada (Consultar código Adjunto)

En las pruebas que has realizado, ¿resulta más adecuado trabajar con una población de mayor tamaño o realizar más iteraciones? Razona la respuesta en base a tus experimentos.

(generations,alpha,mu,lambda,crossoverRate)

(1000,100,25,25,0.95) = 650 (default)

(1000,200,50,50,0.95) = 675.6

(1000,500,100,100,0.95) = 680.4

(2000,100,25,25,0.95) = 632, 650 (lanze 2 veces esta conf con 2 resultados distintos)

(3000,100,25,25,0.95) = 638.0

Como se puede observar contra mas individuos y mas herederos se producen, mayor es la posibilidad de encontrar un individuo superior que si mantenemos la misma población estable y le damos mas tiempo (iteraciones). Es decir que al haber mayor población y actividad de estos por iteración mayor es la posibilidad de que aparezca un superindividuo.

--

2. (2.5 puntos) A partir de los resultados del problema 1, nos hemos dado cuenta de que ciertas combinaciones son más costosas de fabricar y nos gustaría tener en cuenta también el coste, que queremos **minimizar**. De esta forma, vamos a trabajar con una función multi-objetivo en el que, por un lado, queremos maximizar la capacidad de aislamiento y, por otro lado, minimizar el coste del material resultante. Concretamente, existe un coste fijo de 100€ de fabricar el material, que se ve incrementado de la siguiente forma:

- Material que empieza por la capa B o C o D cuesta 20€ más.
- Material que empieza por la capa E o F o G cuesta 50€ más.

Explica el cambio realizado e indica cuál es la mejor solución (combinación de materiales, aislamiento y coste total) que encuentras para este nuevo problema tras la ejecución de 1000 iteraciones. El resto de los parámetros se deja a libertad del alumno (**NOTA:** indicad claramente cuáles son estos parámetros).

NOTA DE IMPLEMENTACIÓN: La clase String tiene un método `startsWith` que nos permite conocer si una cadena comienza por otra subcadena.

NOTA: Para los que **no** han utilizado Opt4J (han implementado el AG desde cero) es recomendable que utilicen una función mono-objetivo que pondere ambos criterios (aislamiento y coste) al 50% cada uno.

```
public class CapasEvaluator implements Evaluator<ArrayList<Integer>>{

    @SuppressWarnings("static-access")
    @Override
    public Objectives evaluate(ArrayList<Integer> fenotipo) {
        // TODO Auto-generated method stub
        double resultado = 0;
        double coste = 100;
        Data d = new Data();
        for (int i = 0; i<fenotipo.size()-1;++i) {
            if(fenotipo.get(0)==1 | fenotipo.get(0)==2 |
fenotipo.get(0)==3) {
                coste = 120;
            }
            if(fenotipo.get(0)==4 | fenotipo.get(0)==5 |
fenotipo.get(0)==6) {
                coste = 150;
            }
            resultado += d.matriz[fenotipo.get(i)][fenotipo.get(i+1)];
            /*AÑADIDO PA EXAMEN*/
            if(i>0 && fenotipo.get(i-1) == 0 && fenotipo.get(i) == 1 &&
(fenotipo.get(i+1) == 3 | fenotipo.get(i+1)== 4 )) {
                resultado = (resultado + (resultado * 0.2));
            }
            if(i>0 && fenotipo.get(i-1) == 5 && fenotipo.get(i) == 7 &&
(fenotipo.get(i+1) == 8 | fenotipo.get(i+1)== 9 )) {
                resultado *= 0;
            }
            /*FIN AÑADIDO PA EXAMEN*/
        }
        Objectives objetivo = new Objectives();
        objetivo.add("Valor Aislante Maximo", Sign.MAX ,resultado);
        objetivo.add("Coste maximo", Sign.MIN ,coste);
    }
}
```

-
3. (2.5 puntos) Queremos extender el problema 2 de forma que se puedan repetir capas. Es decir, puede ocurrir que la misma capa aparezca más de una vez (y otras no aparezcan) pero obteniendo siempre un material de 12 capas. Por cada 2 capas idénticas consecutivas el nivel de aislamiento sumará 20 unidades.

Explica el cambio realizado e indica cómo has definido el genotipo en Opt4J y cuál es la mejor solución (combinación de materiales, aislamiento y coste total) que encuentras para este nuevo problema tras la ejecución de 1000 iteraciones. El resto de los parámetros se deja a libertad del alumno (**NOTA:** indicad claramente cuáles son estos parámetros).

NOTA: Para los que **no** han utilizado Opt4J (han implementado el AG desde cero) es recomendable que utilicen una función mono-objetivo que pondere ambos criterios (aislamiento y coste) al 50% cada uno.

No me da tiempo pero seria cambiar el tipo de genotipo de `PermutationGenotype` ha otro en este caso probaría ha `IntegerGenotype` (uso valores de (0-11) para representar las capas o también se podría usar `DoubleGenotype` pero al ser enteros mejor apostar por el primero