

Técnicas, Entornos y Aplicaciones de Inteligencia Artificial

Práctica 4. ALGORITMOS GENÉTICOS

Objetivo:

utilizar Opt4J para diseñar, resolver y evaluar un problema de optimización mediante AG

Opt4J está disponible en:

Poliformat y en <https://sdarg.github.io/opt4j/index.html>

Opt4J. Entorno libre



A Modular Framework for Meta-heuristic Optimization

Disponible en: <https://sdarg.github.io/opt4j/index.html>

Formulación sencilla de problemas utilizando librerías implementadas en Java

Existe un boletín completo que explica su instalación y uso

Algoritmos Genéticos

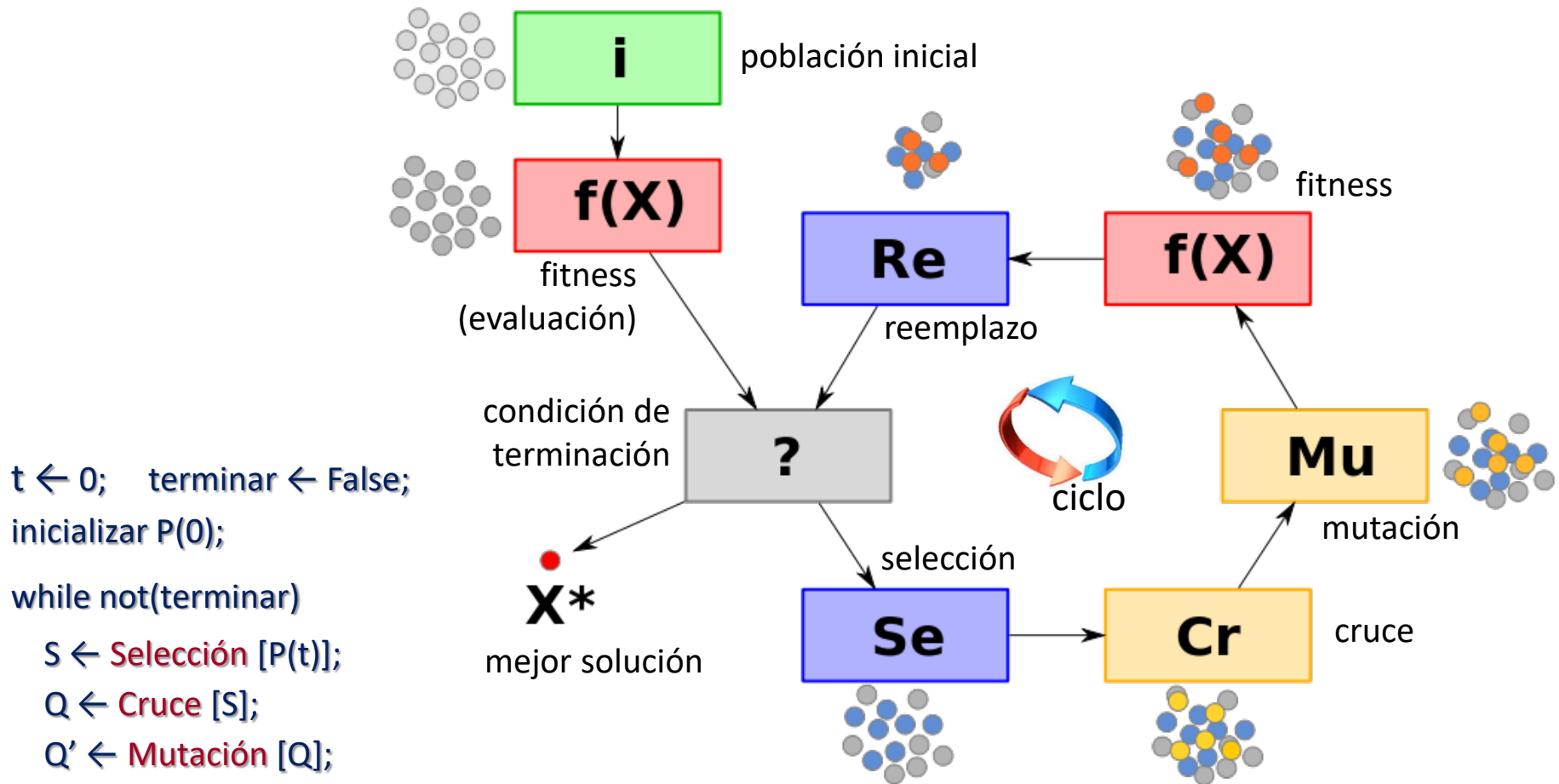
Diseño algoritmo genético

- Diseño del individuo. Codificación y decodificación.
- Función de evaluación (fitness)
- Generación población inicial.
- Selección. Cruce (individuos inválidos). Mutación. Reemplazo.

Evaluación algoritmo genético

- Criterios de evaluación: Fitness versus Soluciones generadas, Tiempo cómputo.
- Tamaños del problema
- Parámetros de evaluación: Población, Selección, Cruce, Mutación, etc.

Práctica 4: Opt4J



$t \leftarrow 0$; terminar \leftarrow False;
inicializar $P(0)$;

while not(terminar)

$S \leftarrow$ Selección [$P(t)$];

$Q \leftarrow$ Cruce [S];

$Q' \leftarrow$ Mutación [Q];

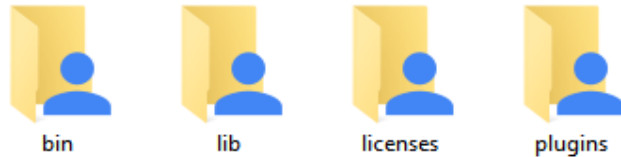
$P(t+1) \leftarrow$ Reemplazo [$P(t)$, S , Q'];

$t \leftarrow t+1$

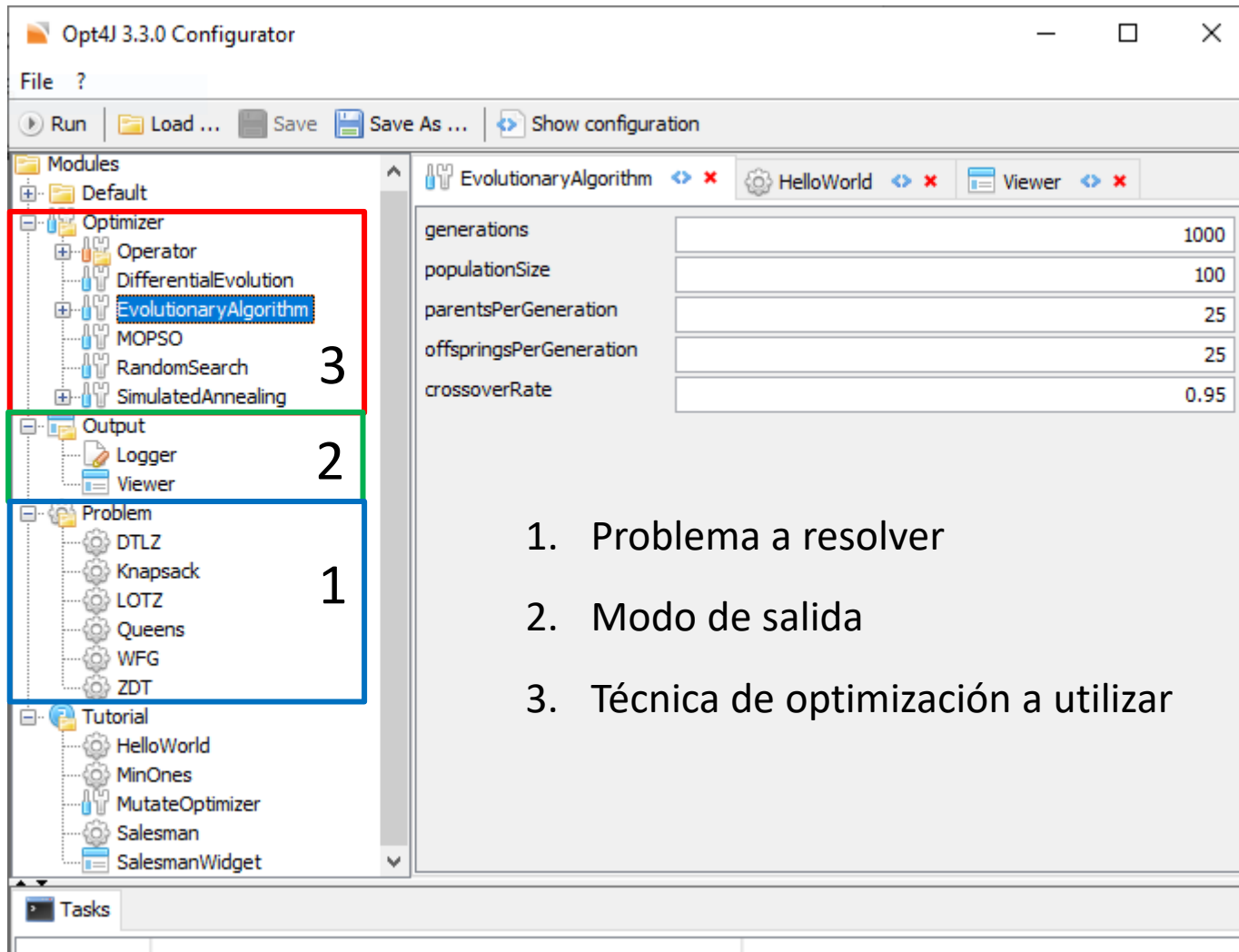
terminar \leftarrow (Convergencia [$P(t+1)$]) OR ($t > \text{límite}$)

end_while

https://commons.wikimedia.org/wiki/File:Evolutionary_algorithm.svg



.../bin/opt4j.bat



Práctica 4: Opt4J

Parametrización del AG

Nº Generaciones/iteraciones

Tamaño población

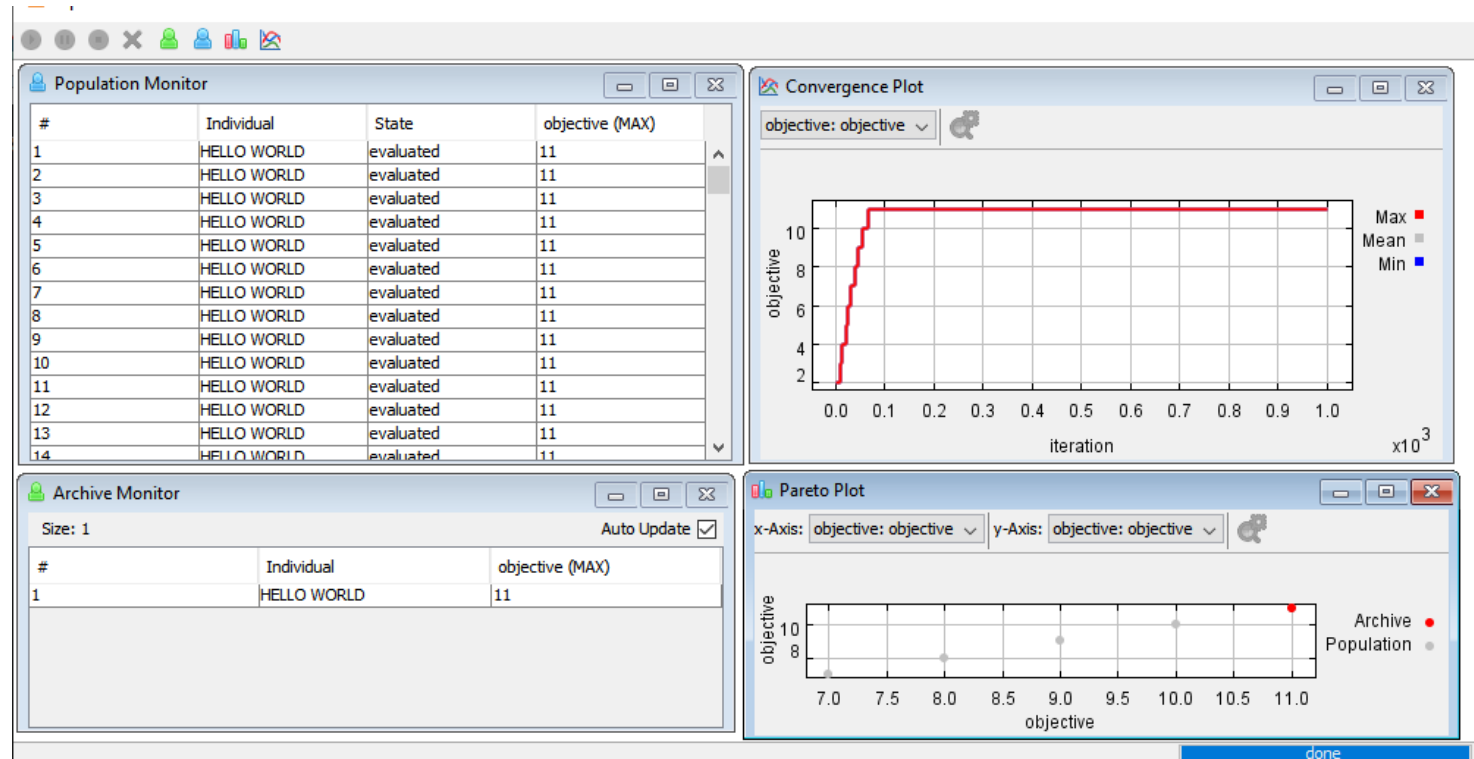
Nº padres seleccionados / Iteración

Nº Hijos generados / Iteración

Probabilidad cruce de dos padres

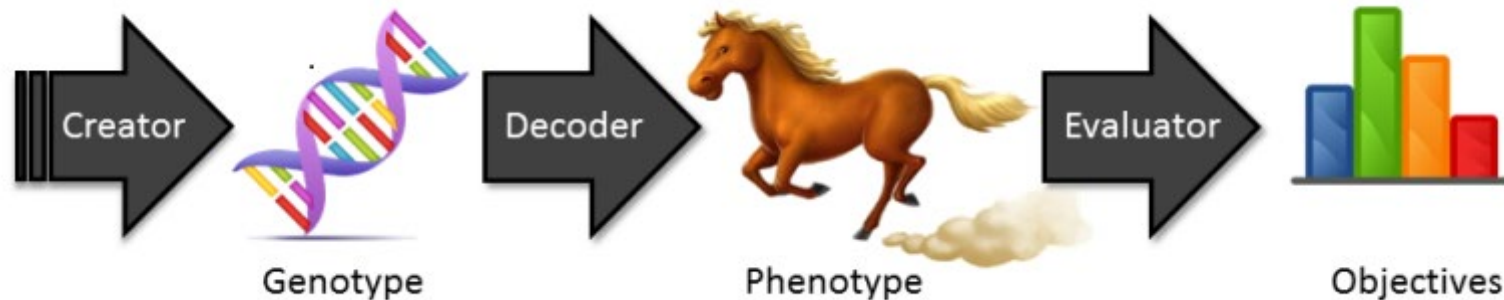
OpsAritmeticas	EvolutionaryAlgorithm	Viewer
generations	1000	
populationSize	100	
parentsPerGeneration	25	
offspringsPerGeneration	25	
crossoverRate	0.95	

Resultados:



- Opt4J permite la importación y resolución de problemas previamente modelados en Java
- Por simplicidad, utilizaremos el **entorno Eclipse**
- ***Configuración de ECLIPSE en Boletín (1.2, 1.3)***
- **RECOMENDABLE:** Importaremos el proyecto-plantilla de Poliformat “***ProyectoAG.zip***” (*Open Projects from File System*) – ver sección 1.3 del boletín
- Modelado del problema en Java (Creator, Decoder y Evaluator)

Básicamente, hay que implementar tres clases



```
public class NombreClaseCreator implements Creator<GENOTIPO>
```

```
public class NombreClaseDecoder implements Decoder<GENOTIPO, FENOTIPO>
```

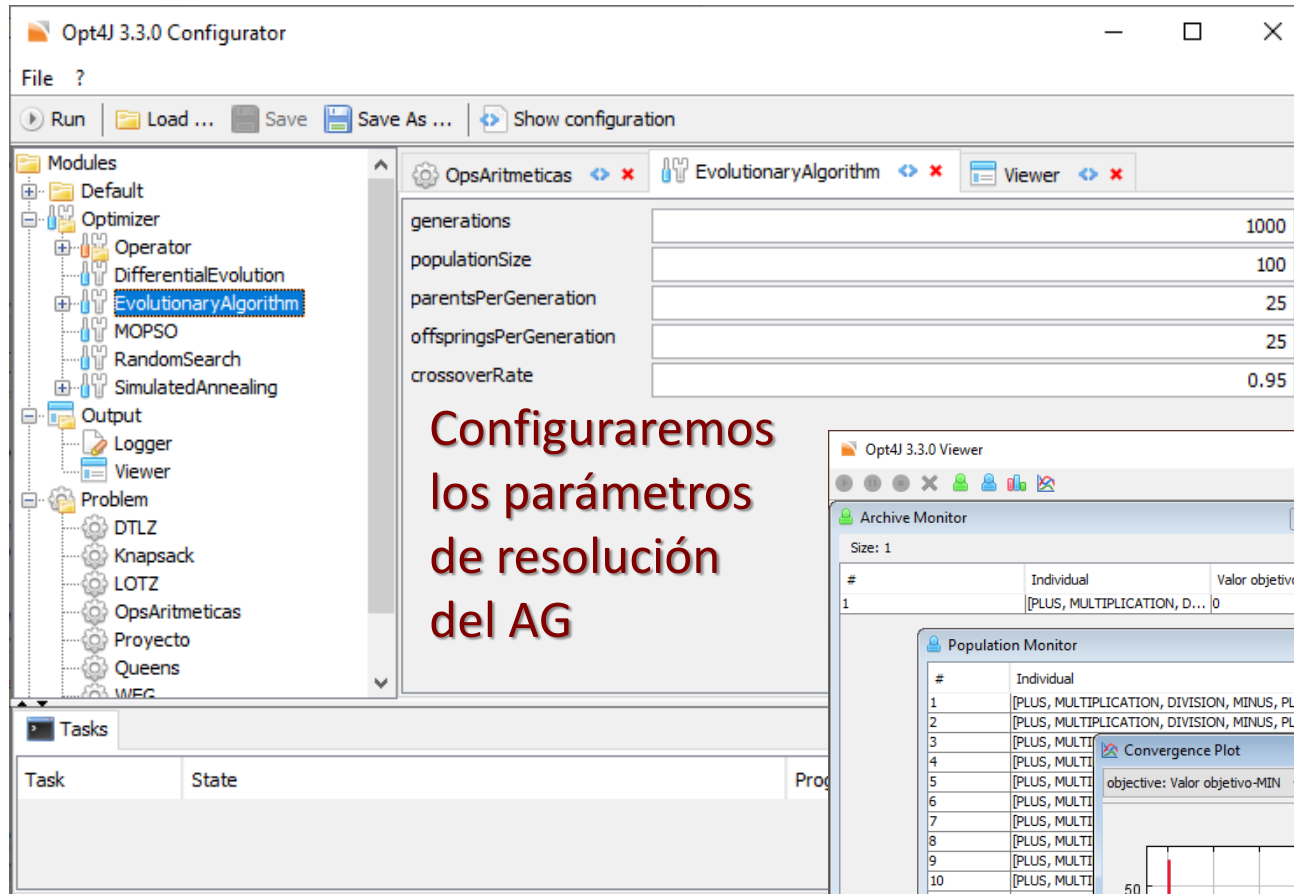
```
public class NombreClaseEvaluator implements Evaluator<FENOTIPO>
```

...más la clase Module que las referencia:

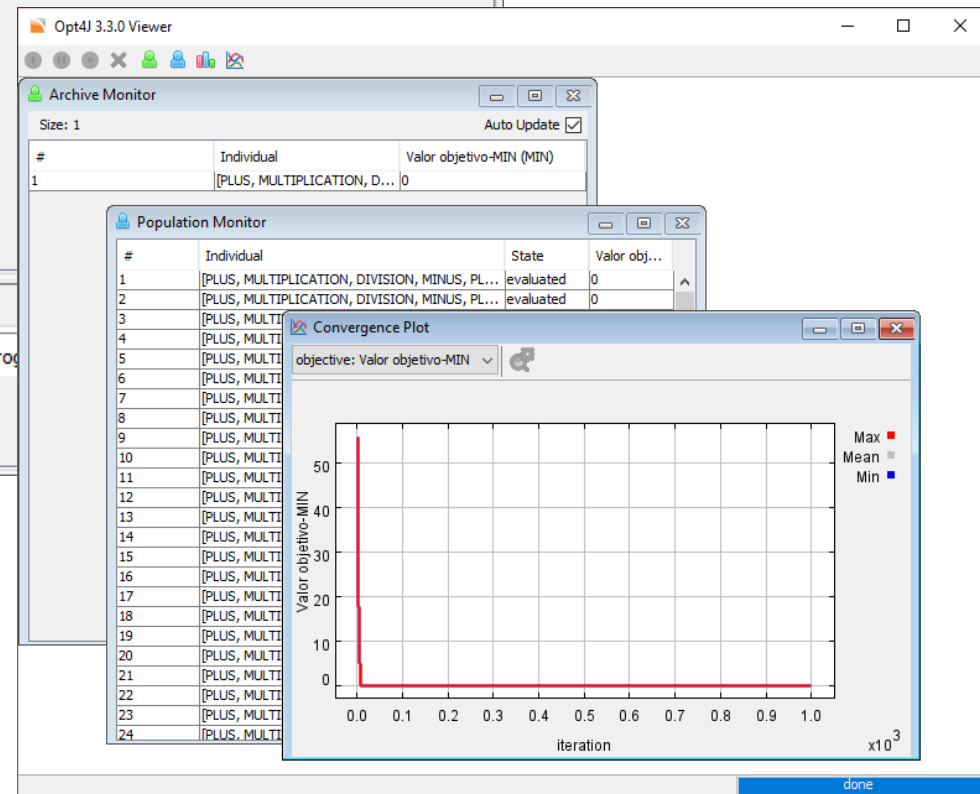
```
public class ClaseModule extends ProblemModule
```

Ver ejemplos en el Boletín

Práctica 4: resolviendo el problema en Opt4J



Configuraremos
los parámetros
de resolución
del AG



Y analizaremos los
resultados

Evaluación:

- Realizar el ejercicio propuesto (se necesitará para el día de la evaluación, en el que se planteará una breve ampliación)

Calendario:

Sem	<u>LABORATORIO</u>	Evaluación
13-XII	Opt4J	
20-XII	Opt4J	
10-I		P4: <i>Aplicac. AG Opt4J</i>

Aplicación y evaluación de Algoritmos Genéticos (15%) P4