

## Técnicas, Entornos y Aplicaciones de Inteligencia Artificial

### Evaluación Práctica-4: Algoritmos Genéticos (Opt4J). 2023-2024.

Nombre: César Martínez Chico 04659582N

#### MUY IMPORTANTE. NORMAS DEL EXAMEN

- 1) Poned el nombre y leed bien cada una de las preguntas.
- 2) Subid a Poliformat todos los ficheros .java correspondientes a la práctica (habrá distintas versiones: versión original y cada uno de los ejercicios). Se puede subir un archivo .zip.
- 3) Contestad a las preguntas siguientes de forma razonada, rellenando los huecos con las respuestas. Se deberá subir también este archivo con las respuestas en formato .RTF o .PDF.
- 4) En cada ejercicio se debe partir de la práctica original. Los ejercicios NO son incrementales.
- 5) ÚNICAMENTE se admitirán los trabajos subidos a la tarea de Poliformat y dentro del plazo de la tarea.
- 6) No está permitida la utilización de aplicaciones de mensajería.

Tiempo: 75 minutos.

**Nota:** todos los ejercicios se deberán realizar con 800 generaciones. El resto de los parámetros se deja a libertad del alumnado.

**NOTA DE IMPLEMENTACIÓN.** Recordad que las colecciones en Java comienzan con el índice 0.

**1. (2.5 puntos, Tiempo estimado: 15')** Utilizando vuestro código original, añadir la siguiente nueva información:

- Existen cuatro nuevas empresas E21, E22, E23 y E24 con el siguiente beneficio esperado (en cientos de €) por unidad de porcentaje (%) de inversión: [1.2, 2.4, 1.5, 2.3]. El nivel de riesgo de estas empresas es: [3, 2, 4, 2].
- Se ha detectado que, si se invierte en las empresas E21 y E22, se incrementa el riesgo total en un 2%.
- Se ha detectado que, si se invierte en las empresas E23 y E24, se decrementa el riesgo total en un 3%.
- Ahora interesa encontrar la distribución de inversiones que **maximiza** el beneficio y **minimiza** la mitad del riesgo.

Indica las modificaciones realizadas y muestra los dos mejores resultados obtenidos en la forma <beneficio,riesgo>:

Lo primero es actualizar los datos, el numero de empresas, beneficios y riesgos.

```

1 package ejercicioEXAMEN;
2
3 public class Data
4 {
5     public static final double minInversionPorEmpresa = 0.0;
6     public static final double maxInversionPorEmpresa = 10.0;
7
8     public static final double maxInversionTotal = 100.0;
9
10    public static final int numEmpresas = 24;
11
12    // beneficio por cada una de las "numEmpresa" empresas
13    public static final double[] beneficioEmpresa =
14    {
15        0.6, 1.24, 1.2, 2.3, 1.2, 3.4, 3.2, 1.5, 3.2, 2.3, 1.7, 3.6, 1.3, 2.15, 1.4, 2.2, 1.5, 2.9, 2.4, 1.2, 1.2, 2.4, 1.5, 2.3
16    };
17
18    // riesgo por cada una de las "numEmpresa" empresas
19    public static final int[] riesgoEmpresa =
20    {
21        1, 3, 2, 3, 1, 7, 6, 2, 5, 5, 2, 6, 3, 7, 6, 3, 2, 4, 5, 5, 3, 2, 4, 2
22    };
23 }

```

Segundo en el evaluador hay que añadir el siguiente código:

Primero los boolean para saber si se ha invertido:

```

//mod ejer 1
boolean invierteEnE21E22 = fenotipo.get(20) > 0 && fenotipo.get(21) > 0;
boolean invierteEnE23E24 = fenotipo.get(22) > 0 && fenotipo.get(23) > 0;

```

Y después cambiar el riesgo total:

```

//mod de ejer 1
if (invierteEnE21E22) {
    riesgo *= 1.02; // Incremento del 2%
}
if (invierteEnE23E24) {
    riesgo *= 0.97; // Decremento del 3%
}

```

Debajo del for donde se calcula el riesgo, evidentemente

Realiza otras pruebas del algoritmo genético modificando los parámetros de "tamaño de la población" e "hijos por generación". De acuerdo a las pruebas que has realizado, ¿resultaría más adecuado trabajar con una población de mayor tamaño o generando más hijos por generación? Razona la respuesta en base a tus experimentos.

Con 1000 iteraciones, 200 de tamaño de población y 30 hijos por generación:

Individual	Valor del beneficio total - MAX: (MAX)	Valor del riesgo total - MIN: (MIN)
[0.0014163975989553554, 0.494533622032408, 8.578483604977098E-8, 9...	27134.17010873352	73.0
[0.0, 1.7730608689349314, 8.688728423258377E-8, 9.987909859926184, ...	27127.173344469513	69.0
[0.011985340958811314, 1.7035542017358036, 0.0, 9.9868532605115, 0.0...	27010.730722325447	68.0
[0.0, 1.697191205382824, 0.0, 9.997612427826077, 0.0, 9.9976163556692...	26960.543454650786	67.0
[0.009089114692842779, 1.7877703614415434, 0.0, 9.776654108696134, ...	26908.302664200208	66.0
[0.0, 1.75955123773523, 0.0, 9.977493038008596, 0.005717511226124701...	26814.205817697097	65.0
[0.23244146060419654, 2.5058640992826495, 0.0, 9.999345843409662, 0....	26777.824390020993	63.0
[3.6253851889793816E-4, 1.669627011111925, 0.0, 9.99971221725216, 0....	26754.25150843515	62.0
[3.324214849556738E-6, 2.234761460976858, 0.0, 10.0, 2.1078750659791...	26581.968275367795	61.0
[8.732684715758212E-5, 1.7903547702564706, 0.0, 10.0, 0.0, 9.99263734...	26570.546467760472	60.0
[0.0, 1.7178463202962617, 0.0, 9.896174826042126, 0.0, 9.924344402150...	26423.004436429306	59.0
[0.0, 2.6824668404191447, 0.0, 9.999485574803485, 0.0343508321994228...	25819.93705483331	58.0
[0.0, 2.738557091728524, 0.0, 9.982791731367909, 0.0, 9.9927748350034...	25741.483014073343	57.0
[0.0015898993686992878, 1.6658265532332075, 0.0, 9.97998978689382, ...	25731.991560974926	55.0
[0.0, 2.8841606247731826, 0.0, 9.985323866415305, 0.2492026755251072...	25713.71680529744	54.0
[0.0, 2.773863147646504, 0.0, 9.985571528084291, 0.1536284263652698, ...	25662.777902473987	51.0
[0.0, 1.66262807417415, 0.0, 9.97124615110767, 0.0021530865602467433...	25347.384033180275	50.0
[0.0, 2.5030955403494723, 0.0, 9.971026346838503, 0.0, 9.988047440522...	25268.342149457312	49.0
[0.0, 2.886850628401999, 0.0, 9.999640983828046, 0.0, 10.0, 9.965560782...	24918.450114257885	48.0
[0.0, 2.6456465049145517, 0.0, 10.0, 9.801966871598477, 9.99993494518...	24855.929293981455	46.0
[0.0, 3.589456172576575, 0.0, 9.999840158799733, 9.759758305542361, 1...	24751.472993481413	44.0
[0.0, 1.7597960749300372, 0.0, 9.969350931118445, 9.76238118439055, 9...	24433.594948930004	43.0
[0.0, 7.707405256417241, 0.0, 9.9999888396553, 9.546381104578034, 9.9...	24000.021087718902	42.0
[0.0, 10.0, 0.0, 9.9996593194368, 9.996496906440356, 9.99684264201899...	23611.187482708854	40.0
[0.0, 2.620236064038114, 9.863638206430041, 10.0, 9.724698921962608, ...	22016.531281859177	39.0
[0.0, 9.466388876968628, 9.442953183976725, 9.259428400881482, 8.984...	21556.722977350604	38.0
[0.19171312353563139, 9.999267794962794, 9.923212019085708, 9.9996...	21201.06340785647	37.0
[0.0, 10.0, 9.860624548001761, 9.999642445729213, 10.0, 9.9999583899...	21175.63506172585	36.0

### Con 1000 iters, 400 de tamaño de población y 30 hijos:

Individual	Valor del beneficio total - MAX: (MAX)	Valor del riesgo total - MIN: (MIN)
[0.02597012167978879, 0.13206599060425395, 0.0, 9.594866832803664, ...	27410.26784826553	67.0
[0.017825741435046526, 0.0, 0.0, 9.938504763775756, 0.04689675974931...	27271.94812774087	64.0
[0.0, 0.0, 0.0, 9.719544467901123, 0.14336042554125625, 9.87341015955...	27255.308609142212	63.0
[1.4541071419644345E-4, 0.0, 0.0, 9.595231752984002, 0.1805475637579...	27140.0884667597	62.0
[0.3336725457411571, 0.0, 0.0, 9.725182251170711, 0.0615845122030858...	27113.047163951887	60.0
[0.0, 0.0, 0.0, 9.991255191854089, 0.0, 9.835433344831644, 9.9754588275...	26968.613404437845	57.0
[0.0014556586603366223, 0.0, 0.0, 9.860786757169796, 0.0037953465934...	26547.806581581324	55.0
[7.140628154694521E-4, 0.0, 0.0, 9.87971953887318, 0.0, 9.87764377195...	25906.5989212565	54.0
[0.001265253357082512, 0.0, 0.0, 9.984013909553964, 0.12486233267417...	25674.45197443366	52.0
[0.0, 0.0, 0.0, 9.8793193051492, 0.0, 9.918006067754757, 9.966845890260...	25543.88051771942	51.0
[0.0, 0.0, 0.0, 9.880604039174832, 0.0, 9.918021518856891, 9.9718594627...	25291.39696531589	49.0
[0.0, 0.0, 0.0, 9.999735434614049, 4.5346608105978765, 9.836921974960...	24934.4759541935	48.0
[0.0, 9.659379035860507, 0.0, 9.725085246858853, 0.0, 9.6929926020037...	24812.420435476004	46.0
[6.181170143596933E-4, 0.0, 0.0, 9.990018270653605, 9.00422447949000...	24703.798311952552	45.0
[1.0407654539763378E-5, 0.0, 4.945723385076151, 9.861742353894392, ...	24703.29289684224	44.0
[0.0, 0.0, 4.93867187726205, 9.86196794523145, 8.886177928762326, 9.9...	24633.376609891235	43.0
[0.0, 0.0, 9.802587539088085, 9.432283014501355, 4.7236132337344445, ...	24133.709608174006	42.0
[0.0, 0.0, 9.538545020931707, 9.702389744342447, 9.086798150167938, 9...	23863.77057994409	41.0
[0.0, 10.0, 0.0, 9.653055635552608, 9.562168971832504, 9.877011413645...	23385.903468206678	40.0
[0.0, 0.0, 9.919982978689143, 9.650003838909068, 9.566078058149385, 9...	23334.921178171615	39.0
[0.0, 8.20176068589748, 9.642636677253519, 9.996629285765984, 9.2115...	21571.586016279354	38.0
[1.410276913716453, 9.789554237119946, 9.343992491507901, 10.0, 9.31...	21090.784020139577	37.0
[0.0, 10.0, 9.967232029390168, 9.961851757638852, 9.719493185238251, ...	21070.047695540812	36.0

### Para 1000 iters, 200 de tamaño y 60 hijos:

Individual	Valor del beneficio total - MAX: (MAX)	Valor del riesgo total - MIN: (MIN)
[0.0, 0.0, 10.0, 9.999621984340068, 9.997477739120452, 9.998242858462...	23575.383441681886	39.0
[1.7441034617828848E-5, 0.0, 0.0, 9.999920396971564, 0.0, 9.999639305...	26868.958583715856	57.0
[8.780649904341422, 0.0, 10.0, 9.999109435741373, 10.0, 9.99996595185...	22518.456409687187	38.0
[0.0, 0.0, 2.3135134067500442, 9.999988720626543, 0.0, 9.999640189686...	25300.344481136635	48.0
[9.981187278261322, 0.0, 9.953937242417513, 9.99939809969993, 9.9996...	20573.72890144667	34.0
[7.749587002548244E-7, 0.0, 0.0, 9.99671364042685, 4.070494708748206...	27179.30412327032	65.0
[0.8501740313854887, 0.0, 10.0, 9.999424506708392, 9.999985834884995...	22028.818611527433	37.0
[0.0, 0.0, 3.29036254255311, 9.999975783418888, 0.0, 9.99574219983769...	24997.538240376936	45.0
[0.0, 0.0, 4.458150087651388, 10.0, 8.824210802679639, 9.997959224407...	24327.929166580274	41.0
[0.0, 0.0, 2.3018411604912745, 10.0, 0.0, 9.999638296175679, 9.95790619...	25833.412092386316	49.0
[1.4386328173141347E-6, 0.0, 3.265041220019876, 9.998945017359548, ...	25017.068834321417	46.0
[0.0789865272931451, 0.0, 9.998362681446391, 9.998996492195545, 10.0...	23576.212959454362	40.0
[0.0, 0.0, 3.203444644297533, 9.999423831161764, 8.562966565418845, ...	24448.840071244715	42.0
[3.106127509719975E-5, 0.0, 0.036832743168739995, 9.99889003510180...	27203.194661551614	70.0
[7.09257210727181, 0.0, 6.566766972200706, 9.999873537388874, 9.3487...	21515.700683631047	36.0
[0.0, 0.0, 0.0, 9.995519100785707, 0.021479968280558204, 9.9878768388...	27168.305835196305	59.0
[0.0, 0.0, 3.259021984598482, 10.0, 8.85933199758242, 9.9963262588485...	24937.732317507347	43.0
[0.0, 0.0, 0.0, 9.944296725097397, 0.0, 9.987802592559234, 9.8955702089...	26469.080675528676	56.0

Como puede observarse, para un riesgo de 41, es mejor la idea de aumentar hijos, ya que para ese riesgo mismo, nos aporta mas beneficios. Pasa lo mismo para riesgo 48 por ejemplo. Aunque las diferencias son pequeñas y creo que con mas iteraciones o simplemente diferentes ejecuciones los resultados podrían variar arbitrariamente por tanto no creo que para este problema en concreto sea super determinante.

2. (3.5 puntos, Tiempo estimado: 25') Si en alguna de las 20 empresas originales se invierte más de un 5% de nuestro capital, nos cobran una determinada comisión (en %) dada por los siguientes valores: [1, 1, 0.5, 2, 1, 1.25, 0.75, 1.2, 1, 2, 1.5, 1.5, 2.1, 1.75, 1.5, 1.75, 1, 1, 0.5, 0.5]. Esto quiere decir que el beneficio de invertir en esa empresa se verá reducido por el valor de la comisión anterior. Por ejemplo, si en E1 se invierte más de un 5%, su beneficio por unidad de porcentaje (%) de inversión ya no será de 0.6, sino  $0.6 - 1\%$  de  $0.6 = 0.594\%$ . Si la cantidad invertida en una empresa no supera al 5%, entonces no habrá que incluir ninguna comisión.

Analizando la evolución de los mercados, se desea incluir la siguiente información adicional:

- En la empresa E3 hay que invertir **obligatoriamente** más de un 5% de nuestro capital.
- La empresa E20 parece prometer altos beneficios, por lo que nos interesa **maximizar** la inversión en la misma.
- La empresa E4 no parece muy solvente, por lo que nos interesa **minimizar** la inversión en la misma.

Indica las modificaciones realizadas y dos de los mejores resultados obtenidos en la forma <beneficio,riesgo,inversión en E20, inversión en E4>:

Primero será añadir el nuevo array de datos al data.java:

```
public class Data
{
    public static final double minInversionPorEmpresa = 0.0;
    public static final double maxInversionPorEmpresa = 10.0;

    public static final double maxInversionTotal = 100.0;

    public static final int numEmpresas = 20;

    // beneficio por cada una de las "numEmpresa" empresas
    public static final double[] beneficioEmpresa =
    {
        0.6, 1.24, 1.2, 2.3, 1.2, 3.4, 3.2, 1.5, 3.2, 2.3, 1.7, 3.6, 1.3, 2.15, 1.4, 2.2, 1.5, 2.9, 2.4, 1.2
    };
    public static final double[] comisiones = {1, 1, 0.5, 2, 1, 1.25, 0.75, 1.2, 1, 2, 1.5, 1.5, 2.1, 1.75, 1.5, 1.75, 1, 1, 0.5, 0.5};
    // riesgo por cada una de las "numEmpresa" empresas
    public static final int[] riesgoEmpresa =
    {
        1, 3, 2, 3, 1, 7, 6, 2, 5, 5, 2, 6, 3, 7, 6, 3, 2, 4, 5, 5
    };
}
```

Una vez hecho esto, lo siguiente será modificar el bucle del evaluador para comprobar si se invierte un 5% o más y así aplicar la comisión de la siguiente forma:

```
for (int i = 0; i < fenotipo.size(); i++) {
    double inversion = fenotipo.get(i);
    double comision = 0;

    // Aplicar comisión si inversion -> + 5%
    if (i < 20 && inversion > 5) {
        comision = Data.comisiones[i] * Data.beneficioEmpresa[i] / 100;
    } //simplemente aqui calculamos lo que hay que restar al beneficio final

    if (inversion > 0) {
        beneficio += inversion * (Data.beneficioEmpresa[i] - comision); //restamos comision
        riesgo += Data.riesgoEmpresa[i];
    }
}
```

- Ahora lo que debemos hacer para comprobar que “En la empresa E3 hay que invertir **obligatoriamente** más de un 5% de nuestro capital.” Lo que he hecho ha sido añadir una variable que se ponga a true si invertimos menos del 5%, entonces si esa variable está a true, al final descartamos la solución, si no, significa que hemos invertido 5 o más por ciento en la E3 he añadido la variable, el if y modificado el for de evaluador como sigue:

```

boolean inversionEnE3 = false; //aquí guardo si se invierte o no el 5% en la 3

for (int i = 0; i < fenotipo.size(); i++) {
    double inversion = fenotipo.get(i);
    double comision = 0;

    if(i == 3 && inversion < 5) {
        inversionEnE3 = true; //si invertimos menos del 5% entonces true
    }
    // Aplicar comisión si inversion -> + 5%
    if (i < 20 && inversion > 5) {
        comision = Data.comisiones[i] * Data.beneficioEmpresa[i] / 100;
    } //simplemente aquí calculamos lo que hay que restar al beneficio final

    if (inversion > 0) {
        beneficio += inversion * (Data.beneficioEmpresa[i] - comision); //restamos comision
        riesgo += Data.riesgoEmpresa[i];
    }
}

if(inversionEnE3) { //hacemos que no se elija esta solucion si no cumple las restricciones del ejercicio
    beneficio = Double.MIN_VALUE;
    riesgo = Double.MAX_VALUE;
}

```

- Para cumplir ahora que “La empresa E20 parece prometer altos beneficios, por lo que nos interesa **maximizar** la inversión en la misma.” Y “La empresa E4 no parece muy solvente, por lo que nos interesa **minimizar** la inversión en la misma.” Lo que voy a hacer primero es añadir variables y modificar el for para saber cuanto invierto en las empresas 20 y 4:

```

double inversionEnE20 = 0.0;
double inversionEnE4 = 0.0;

boolean inversionEnE3 = false; //aquí guardo si se invierte o no el 5% en la 3

for (int i = 0; i < fenotipo.size(); i++) {
    double inversion = fenotipo.get(i);
    double comision = 0;

    if(i == 20) {
        inversionEnE20 = inversion;
    }

    if(i == 4) {
        inversionEnE4 = inversion;
    }
}

```

De esta forma me guardo lo que estoy invirtiendo en 20 y 4 de forma rápida para acceder a esos valores en los objetivos, y ahora voy a modificar los objetivos del problema, de forma que maximice y minimice respectivamente E20 y E4:

```

Objectives objectives = new Objectives();
objectives.add("Valor del beneficio total - MAX: ", Sign.MAX, beneficio * 100); //por 100 porque es en cientos de
objectives.add("Valor del riesgo total - MIN: ", Sign.MIN, riesgo);

objectives.add("Inversion en E20 - MAX", Sign.MAX, inversionEnE20 );
objectives.add("Inversion en E4 - MIN: ", Sign.MIN, inversionEnE4);

```

Ahora le estoy diciendo al solver que tenga en cuenta que debo maximizar la inversión en la empresa 20 y minimizar la inversión en la empresa 4.

3. (3 puntos, Tiempo estimado: 20') Las empresas originales E1 y E2 tienen la intención de fusionarse en una nueva empresa E21. Aunque sigue siendo posible invertir en ambas empresas por separado, la cantidad que se invierta en E1 o E2 se deberá invertir igualmente en E21. Por ejemplo, si en E1 se invierte un 2.5% y en E2 se invierte un 5%, la cantidad que se deberá invertir obligatoriamente en E21 es de 2.5+5=7.5%. El valor de beneficio y riesgo de E21 es 3.5 y 6, respectivamente.

Indica las modificaciones realizadas y qué harías para asegurar que todos los individuos sean soluciones válidas con respecto a E21, sin tener que modificar el valor del fitness. Indica cuál es el mejor resultado (con respecto al beneficio total) que has obtenido en la forma <beneficio, riesgo, inversión en E1, inversión en E2, inversión en E21>

Lo primero cambiar la data:

```
1 package ejercicioEXAMEN;
2
3 public class Data
4 {
5     public static final double minInversionPorEmpresa = 0.0;
6     public static final double maxInversionPorEmpresa = 10.0;
7
8     public static final double maxInversionTotal = 100.0;
9
10    public static final int numEmpresas = 21;
11
12    // beneficio por cada una de las "numEmpresa" empresas
13    public static final double[] beneficioEmpresa =
14    {
15        0.6, 1.24, 1.2, 2.3, 1.2, 3.4, 3.2, 1.5, 3.2, 2.3, 1.7, 3.6, 1.3, 2.15, 1.4, 2.2, 1.5, 2.9, 2.4, 1.2
16    };
17
18    // riesgo por cada una de las "numEmpresa" empresas
19    public static final int[] riesgoEmpresa =
20    {
21        1, 3, 2, 3, 1, 7, 6, 2, 5, 5, 2, 6, 3, 7, 6, 3, 2, 4, 5, 5
22    };
23 }
```

Luego añadiría en el decoder un if de forma que se deba invertir si o si la suma:

```
1 package ejercicioEXAMEN;
2
3 import java.util.ArrayList;
4
5 public class inversionesDecoder implements Decoder<DoubleGenotype, ArrayList<Double>> {
6     @Override
7     public ArrayList<Double> decode(DoubleGenotype genotipo) {
8         ArrayList<Double> fenotipo = new ArrayList<Double>();
9         double inversionTotal = 0.0;
10        for (int i = 0; i < genotipo.size(); i++) {
11            double inversionEmpresa = genotipo.get(i);
12            if (inversionTotal + inversionEmpresa <= Data.maxInversionTotal) {
13                inversionTotal += inversionEmpresa;
14                fenotipo.add(genotipo.get(i));
15            } else {
16                fenotipo.add(0.0);
17            }
18            if(i == 21) {
19                fenotipo.add(fenotipo.get(1) + fenotipo.get(2));
20            }
21        }
22        return fenotipo;
23    }
24 }
```

4. (1 punto, Tiempo estimado: 5') Explica (no es necesario implementar nada) cuál sería el mejor genotipo si simplemente se deseara obtener una solución que indicara si se debe invertir en una empresa o no. ¿Y cuál sería el mejor genotipo si queremos mantener una información del tipo "invertir", "invertir-con-incertidumbre" "no-invertir" y "desconocido" para cada una de las empresas? Razona las respuestas.

1) En el primer caso, considero que es mejor usar un *BooleanGenotype* que es una secuencia de valores T/F (puede contener repeticiones) ya que nos sirve simplemente para guardar TRUE o FALSE para cada empresa, refiriéndonos con ese

---

boolean a si invertimos o no en dicha empresa.

2) En el segundo caso, optaremos por *SelectGenotype<V>*, siendo el tipo V de la forma:

[“invertir”, “invertir-con-incertidumbre”, “no-invertir”, “desconocido”]

ya que nos permite elegir CON REPETICION entre las diferentes 4 opciones del tipo V para cada empresa. Por tanto, podemos decidir si invertir o no en una empresa, o si invertir con incertidumbre, o desconocido si (imagino) aun no sabemos que hacer con esa empresa.