

# Algoritmos genéticos: Aplicación al problema de la mochila.

Javier Hidalgo Sánchez  
Universidad Carlos III de Madrid  
Coria, Cáceres, España

100056426@alumnos.uc3m.es

Jesús Ignacio Turrado Martínez  
Universidad Carlos III de Madrid  
León, España

100056173@alumnos.uc3m.es

## ABSTRACT

En los años 70, surgió una de las líneas más prometedoras de la inteligencia artificial, los algoritmos genéticos. Reciben este nombre, porque se inspiran en la evolución biológica. Realizaremos una descripción del Algoritmo Genético Simple, que es la base de los algoritmos genéticos. Posteriormente veremos una serie de mejoras u extensiones del AGS para adaptar el algoritmo a cada tipo de problema. Para finalizar, ilustraremos la teoría con un ejemplo: El problema de la Mochila. A pesar de ser un problema relativamente simple, esta situación se presenta con cierta frecuencia en los ámbitos económico e industrial, donde la mochila suele representar la restricción presupuestaria (cantidad máxima de recursos económicos de los que se dispone) y donde la utilidad de los objetos seleccionados se equipara a un beneficio económico por adquirir o llevar a cabo ciertas acciones.

**Keywords:** algoritmo genético, problema de la mochila.

## 1 INTRODUCCIÓN

Un algoritmo es una serie de pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico. En los años 1970, de la mano de John Henry Holland, surgió una de las líneas más prometedoras de la inteligencia artificial, la de los **algoritmos genéticos**. Son llamados así porque se inspiran en la evolución biológica y su base genético-molecular.

Por lo tanto, son técnicas de resolución de problemas de optimización, que se basan en emular la HERENCIA GENÉTICA y el PRINCIPIO DE SUPERVIVENCIA.

En la naturaleza los individuos de una población compiten entre sí en la búsqueda de recursos tales como comida, agua y refugio. Incluso los miembros de una misma especie compiten a menudo en la búsqueda de un compañero. Aquellos individuos que tienen más éxito en sobrevivir y en atraer compañeros tienen mayor probabilidad de generar un gran número de descendientes. Por el contrario, individuos poco dotados producirán un menor número de descendientes. Esto significa que los genes de los individuos mejor adaptados se

propagarán en sucesivas generaciones hacia un número de individuos creciente. La combinación de buenas características procedentes de diferentes predecesores, puede a veces producir "superindividuos", cuya adaptación es mucho mayor que la de cualquiera de sus antecesores. De esta manera, las especies evolucionan logrando unas características cada vez mejor adaptadas al entorno en el que viven.

Los Algoritmos Genéticos son adecuados en aquellos problemas para los cuales no existen técnicas especializadas. Incluso en el caso en que dichas técnicas existan, y funcionen bien, pueden efectuarse mejoras de las mismas combinándolas con los Algoritmos Genéticos.

## 2 ALGORITMO GENÉTICO SIMPLE

### 2.1 ALGORITMO

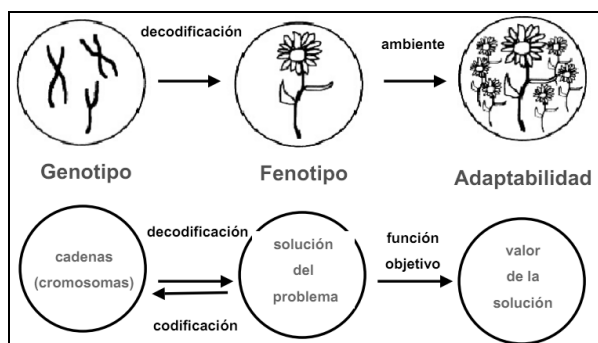
El algoritmo genético simple (AGS) es de la siguiente manera:

- $t=0$ ;
- Crear población inicial  $\rightarrow P(t)$ ;
- Evaluar  $P(t)$ ;
- Repetir hasta condición de finalización
  - $t = t + 1$ ;
  - Selección  $P(t-1) \rightarrow P(t)$ ;
  - Cruce  $P(t)$ ;
  - Mutación  $P(t)$ ;
  - Evaluar  $P(t)$ ;

Siendo  $t$  la variable temporal y  $P(t)$  la población en cada instante de tiempo.

### 2.2 REPRESENTACIÓN DE VARIABLES

La representación de variables en algoritmos genéticos guarda un paralelismo con la genética: cada SOLUCIÓN del problema (FENOTIPO) se representa por un INDIVIDUO de la población (GENOTIPO).



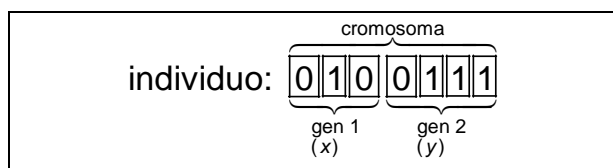
**Figura 2.1. Paralelismo con la genética.**

Los individuos (o genotipos) están formados por una cadena de caracteres o CROMOSOMAS.

Los cromosomas están formados por GENES o caracteres que representan la codificación en binario de las VARIABLES de optimización. A veces se identifica un GEN con cada carácter y otras veces se identifica un gen con el conjunto de caracteres que representan una variable.

Los genes pueden tomar dos diferentes VALORES (0, 1), que se denominan ALELOS.

Supongamos que tenemos un problema en el que hay dos variables  $x, y \in \mathbb{Z}$  cuyos valores están acotados:  $0 \leq x \leq 7, 0 \leq y \leq 15$ . Observamos que la variable  $x$  puede tomar ocho valores diferentes, y la  $y$  puede tomar dieciséis. Esto significa que para codificar la variable  $x$  necesitamos un gen de tres alelos ( $2^m \geq 8 \rightarrow m = 3$ ) y para la variable  $y$  necesitamos cuatro ( $2^m \geq 14 \rightarrow m = 4$ ). Tendremos entonces un cromosoma de siete alelos formado por dos genes. Si  $x = 2$  e  $y = 7$ , la representación en binario del individuo es la representada en la Figura 2.2.



**Figura 2.2. Representación en binario.**

## 2.3 POBLACIÓN INICIAL

El objetivo de establecer una población inicial es el de seleccionar los puntos de partida de la búsqueda en el espacio de soluciones.

La definición del tamaño de la población es muy importante: un valor muy grande provocará un consumo mayor de recursos y un valor muy pequeño barrerá un espacio de búsqueda menor y, por ello, tardará más en encontrar la mejor solución.

La población inicial en el algoritmo genético simple (AGS) se genera de forma aleatoria.

## 2.4 EVALUACIÓN

La evaluación consiste en asignar una SALUD a los individuos de la población. Lo más sencillo, si se puede, es utilizar el valor de la función objetivo.

La función objetivo (o función de adaptación) para cada cromosoma devuelve un número real, que se supone es proporcional a la adaptación del individuo al problema.

En la sección 3 se verán diferentes métodos de adaptación de la función objetivo a cada tipo de problema.

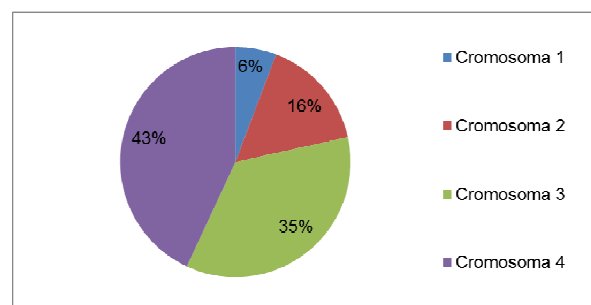
## 2.5 SELECCIÓN

La selección determina los individuos que van a ser “padres” de la siguiente generación. Es un intento de imitación de la SELECCIÓN NATURAL, es decir, supervivencia del individuo más sano. En nuestro caso, los individuos más sanos son aquellos que tienen mejor salud.

El método de selección consiste en tomar tantos padres como el tamaño de la población y asignando a cada individuo una salud relativa de acuerdo a su contribución a la salud total de la población. Es el conocido MÉTODO DE LA RULETA.

$$P_i = \frac{s_i}{\sum_{j=1}^N s_j} \quad (0.1)$$

Tomemos como ejemplo la Figura 2.3: suponemos que tenemos cuatro cromosomas. En la ruleta representamos su salud relativa. Como observamos, el cromosoma 4 es el que tiene mayor probabilidad de ser padre de la siguiente generación al ser el individuo más apto. El caso contrario es el cromosoma 1.



**Figura 2.3. Método de la Ruleta**

## 2.6 CRUCE

El cruce consiste en combinar las cadenas de caracteres de los padres por parejas y obtener su descendencia, imitando la HERENCIA GENÉTICA, es decir, el “hijo” tendrá características de los padres.

El método consiste en tres pasos:

- Emparejamiento: se toman los padres de dos en dos aleatoriamente, por ejemplo, de acuerdo al orden de la selección).
- Probabilidad de cruce: de acuerdo a una determinada probabilidad, los padres se cruzan o pasan a la siguiente generación sin cruce, es decir, no tienen descendencia.
- Cruce: se utiliza el CRUCE SIMPLE, en el que se selecciona aleatoriamente una posición del cromosoma a partir de la cual los dos padres intercambian sus valores.

Para ilustrar el CRUCE SIMPLE vemos la Figura 2.4. Se ha elegido aleatoriamente que los padres crucen por la posición dos. El hijo tiene los dos primeros genes del padre y los tres siguientes son de la madre. La hija tiene los dos primeros de la madre y los tres siguientes del padre.

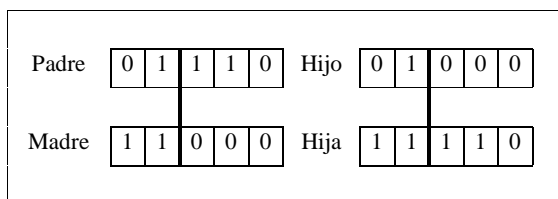


Figura 2.4. Cruce simple.

## 2.7 MUTACIÓN

La mutación tiene como objetivo añadir combinaciones genéticas para favorecer la expansión por el espacio de búsqueda y evitar óptimos locales. Continuando con nuestro paralelismo con la naturaleza, la mutación es una imitación de la MUTACIÓN GENÉTICA.

El método consiste en cambiar, de acuerdo a una tasa, uno de los caracteres binarios por su contrario. La tasa de mutación suele ser muy baja para no perder el efecto de la supervivencia del más sano.

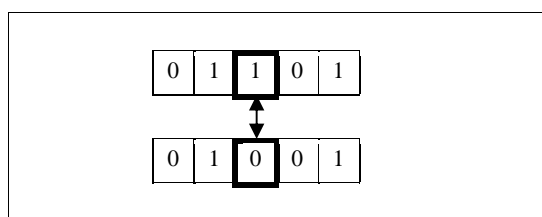


Figura 2.5. Mutación.

## 3 DISEÑO DE ALGORITMOS GENÉTICOS

### 3.1 DEFINIR VARIABLES Y FUNCIÓN OBJETIVO

Hay que tener en cuenta las indicaciones genéricas para cualquier problema de optimización. Además, si es posible, intentar acotar el espacio de soluciones.

Para definir las variables:

- Formarán parte de los cromosomas  $\text{Cromosoma} = [v_1, v_2, \dots, v_n]$ .
- Intentar no alargar los cromosomas más de lo necesario, ya que eso tiene un coste.
- Determinar si hay un rango posible de valores.

Para definir la función objetivo:

- A veces la definición es clara por la naturaleza del problema.
- Otras veces es compleja su definición (por ejemplo: trayectoria de robots) y hay alternativas.
- Problema del multiobjetivo.

### 3.2 REPRESENTACIÓN DE VARIABLES

Para la discretización de las variables continuas, hay que determinar la exactitud:

- Número de intervalos:  $2^m - 1$  ( $m$  = número de bits).
- Rango de la variable:  $[V_{\max}, V_{\min}]$
- Exactitud o longitud del intervalo =  $(V_{\max} - V_{\min}) / (2^m - 1)$ .

Para la codificación en binario:

- Cada intervalo de valores de la variable se representa con un único valor en binario.
- $V = V_{\min} + X_b \cdot (V_{\max} - V_{\min}) / (2^m - 1)$ .
- Variantes en el número de intervalos =  $2^m$  (extremo inferior, extremo superior, centrada).

Otras codificaciones:

- Reales.
- Específicas del problema. Caso especial en problemas de permutación, que veremos más adelante.

### 3.3 POBLACIÓN INICIAL

Lo primero es fijar el parámetro del tamaño de la población.

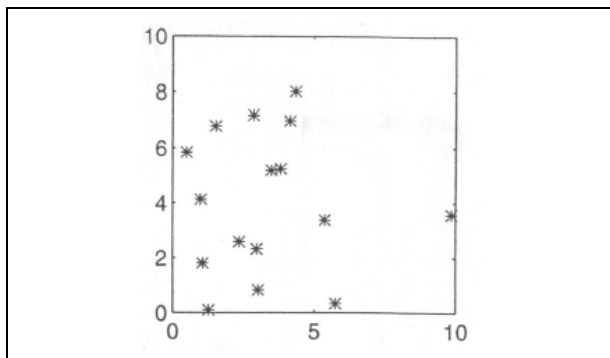
Los tipos de tamaño, pueden ser:

- Tamaño grande:

- Buena muestra de partida para rastrear el espacio de búsqueda.
  - En general, más tiempo para llegar a la solución óptima.
- Tamaño pequeño:
  - Peligro de convergencia rápida a un óptimo local.
  - Más rápido
- No obstante, hay que estudiar el comportamiento específico de CADA PROBLEMA.
- En ocasiones, se toma una población inicial con tamaño superior a la población de las sucesivas generaciones.

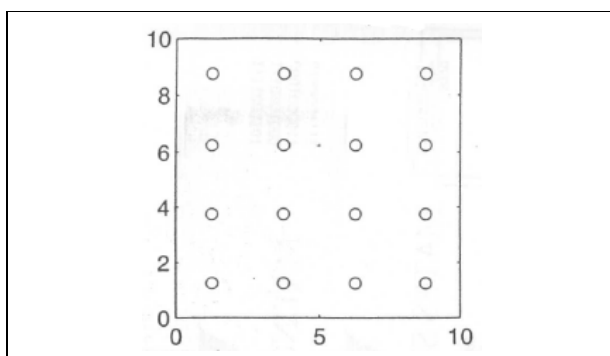
Para generar la muestra, se puede proceder de las siguientes maneras:

- Al azar: como en el AGS. Puede salir no representativa (50% de 0, y 50% de 1).



**Figura 3.1. Muestra aleatoria.**

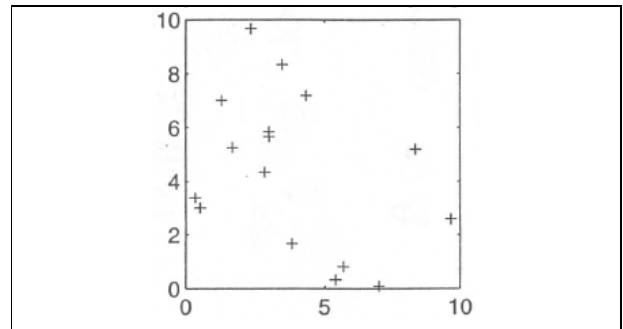
- Uniforme: individuos barriendo uniformemente el espacio de soluciones (fenotipos).



**Figura 3.2. Muestra uniforme.**

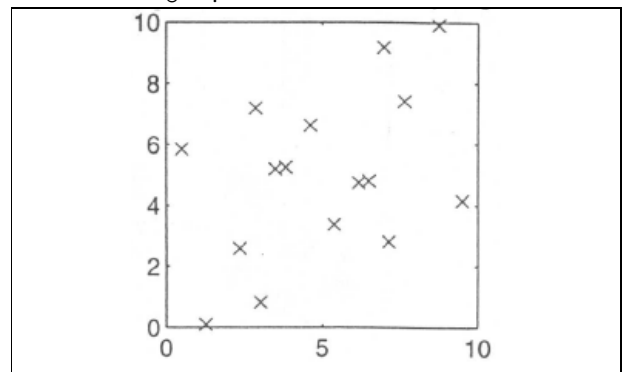
- Aleatoria parcial:
  - Se divide el tamaño de la muestra entre el número de parámetros.
  - Se hacen k submuestras, en las que uno de los parámetros toma valores uniformemente repartidos y los otros valores aleatorios.

- Compromiso entre asegurar una muestra repartida y estocasticidad.



**Figura 3.3. Muestra aleatoria parcial.**

- Invertida:
  - La mitad de la muestra al azar y la otra mitad por inversión binaria.
  - Uniformidad en binario. Se asegura contar inicialmente con todo el material genético y a partes iguales
  - .



**Figura 3.4. Muestra invertida.**

- Dependencia del problema y en gran medida del TAMAÑO.

### 3.4 EVALUACIÓN

Problemas que pueden surgir en la función objetivo:

- Puede dar valores negativos.
- Puede ser no discriminatoria, esto significa que todos los individuos estén en un rango muy parecido.
- Que el problema sea de minimización.

Las soluciones genéricas a estos tres problemas pueden ser:

- Escalado:
  - Traslación:  $f' = f + b$  trasladamos el cero a la cota inferior, y nos resuelve el problema de minimización. Esto nos puede meter en el problema de que NO sea discriminatorio.

- Escalado lineal  $f' = a \cdot f + b$ .
- Escalado potencial  $f' = f^k$ .
- Ranking.

Las soluciones específicas para el problema de minimización pueden ser:

- Cota Superior  $f' = CotaSup.f$ .
- Inversa  $f' = \frac{1}{f}$ .

### 3.5 SELECCIÓN

Independientemente del método de selección, una posibilidad consiste en reducir los aspirantes a padres a los mejores:

- Padres + hijos: Los (N - n) peores mueren. Los n mejores son candidatos a padres y componen con sus (N - n) hijos la siguiente generación.
- Umbral de supervivencia: Sobreviven los que lo superan. El problema es la determinación de este umbral y su evolución, ya que si es fijo, al principio no pasa ninguno y al mejorar la población lo pasará la mayoría. También se produce un número variable de individuos que habrá que compensar.

Hay que determinar cuántos hijos va a tener cada individuo de la población (toda o la parte que haya superado la selección previa):

- Todos un hijo: No representa la ida de la “supervivencia del más apto”, pero puede tener sentido con selección natural previa. Es fácil de implementar.
- Ruleta: Como el AGS.
- Valor esperado: Se calculan las saludes normalizadas y el número de descendientes esperado. Se asignan las partes enteras. El resto:
  - Reparto determinista: por orden del valor fraccionario.
  - Reparto por ruleta: probabilidad proporcional a la fracción.
  - Reparto estocástico: orden y probabilidad igual a la fracción.
- Torneo: Se toman aleatoriamente conjuntos de k padres posibles. Los ganadores (mejores) serán los padres.
- Elitismo: Este concepto expresa la idea de que el mejor individuo de la actual generación pase sin modificar a la siguiente generación. De esta forma no se perderá el mejor cromosoma. Al resto de la población se le aplica la reproducción normalmente. Suelen ser algoritmos que convergen más rápido a la solución, aunque esto dependerá de cada problema.

### 3.6 CRUCE

Tiene tres etapas:

- Emparejamiento, que puede ser:
  - Por orden: se ordenan todos los posibles padres de acuerdo a su salud y se forman parejas por orden. A veces se opta por el orden contrario. Muy simple y fácil de implementar, típicamente combinado con selección natural previa y “todos un hijo”.
  - Al azar: como en AGS
- Probabilidad de cruce: se cruza una parte de los emparejados, de acuerdo a una probabilidad  $P_c$  (parámetro a determinar).
- Cruce, que puede ser:
  - Cruce simple: Por un punto determinado al azar.
  - Cruce múltiple: Por n puntos determinados al azar.
  - Cruce uniforme: Bit a Bit con un 50%. Máscara.

### 3.7 MUTACIÓN

A una pequeña parte de la población, se le introducen cambios en su material genético.

La tasa a la que se introducen estos cambios, puede ser:

- Tasa constante: de acuerdo a una probabilidad  $P_m$ .
- Tasa decreciente: se parte de una  $P_m$  relativamente alta y rebajarla generación a generación (con exponencial, potencias decrecientes de 2, lineal,...). Esto favorece primero la exploración y luego la convergencia.
- Tasa variable en función de la posición del Bit: favorecer las mutaciones para ajustar la solución al óptimo o para dar grandes saltos. Lo normal es que los Bits significativos los determine la selección.
- Operador Inversión: invertir un intervalo entero determinado al azar.

### 3.8 TEST DE FINALIZACIÓN

Para realizar el test de finalización, se requiere definir antes una serie de criterios, como son:

- Número de generaciones: ligado al tiempo del algoritmo.
- Número de evaluaciones: compromiso entre tamaño de la población y número de generaciones, es decir, si yo aumento el número de generaciones para un mismo tipo de evaluación, tendré que disminuir el número de individuos.
- Número de generaciones sin mejora: existe el peligro de atravesar una fase larga en un óptimo local.
- Acercamiento a un valor conocido de antemano: cuando hay un valor objetivo satisfactorio, y se busca principalmente la composición de valores de las variables. Es decir, si no sé cuál es el óptimo, pero sí un valor que me sirve (solución factible), me quedo con ella.

## 4 ALGORITMOS GENÉTICOS CON RESTRICCIONES

Los algoritmos genéticos se inventaron para poder resolver cualquier problema de optimización. Sin embargo, no se suelen “llevar bien” en los problemas con restricciones. Para ello, existen soluciones standart a los problemas que presentan las restricciones.

Las restricciones dividen el espacio de búsqueda en zonas factibles y no factibles. Las cuestiones que surgen al resolver problemas mediante algoritmos genéticos son varias:

- ¿Cómo evaluar los individuos no factibles?
- ¿Cómo relacionar los individuos factibles y los no factibles?
- ¿Son validos los operadores de selección, cruce y mutación que hemos diseñado?

Por ejemplo, sea el espacio de búsqueda representado en la Figura 4.1, pero, por restricciones del problema, tan sólo las zonas rayadas son posibles soluciones factibles. El problema aparece cuando un individuo pertenece a la región no factible, como el que está señalado.

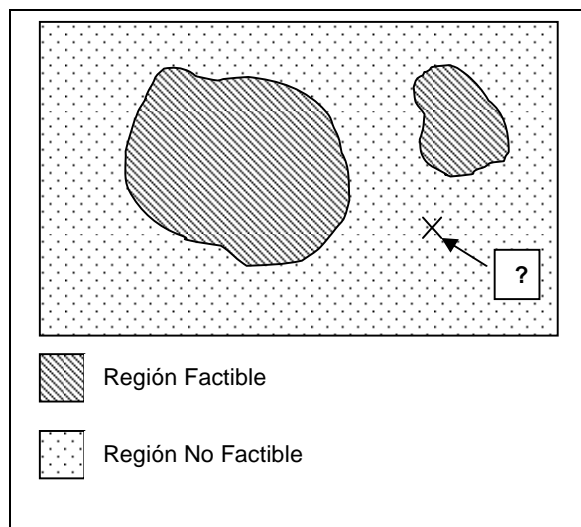


Figura 4.1. Espacio de soluciones con restricciones.

Vamos a ver dos métodos para solucionar los problemas de individuos no factibles.

### 4.1 PENALIZACIONES

Consiste en añadir a la función objetivo penalizaciones que se aplican cuando se violan las restricciones.

Las penalizaciones suelen medir el “grado de infactibilidad” del individuo.

Existen varios tipos de penalizaciones:

- Penalizaciones estáticas: no varían con el avance del algoritmo. El valor de la penalización permanece constante generación tras generación.
- Penalizaciones dinámicas: varían de generación en generación. Normalmente, se penaliza menos al principio del algoritmo para favorecer que los individuos se agrupen en las zonas buenas (factibles). Según avanza el tiempo, la penalización crece de valor para “matar” las soluciones malas (o no factibles).

$$\frac{y}{G} = \frac{\text{generación actual}}{\text{nº de generaciones a ejecutar}} \quad (0.2)$$

$y = 1, 2, \dots$

- Penalización mortal: consiste en eliminar a todos los no factibles. Es una técnica peligrosa cuando la región de no factibles es grande.
- Superioridad factible: se basa en el principio de que cualquier factible es mejor que cualquier no factible.

## 4.2 OPERADORES ESPECÍFICOS

Suele ocurrir que al cruzar individuos factibles resulte uno no factible. Para ello realizamos “trucos” que aseguren que el nuevo individuo sea factible. Estos trucos consisten en definir los operadores de cruce y mutación de forma que arrojen soluciones factibles.

### 4.2.1 CRUCE PMX

Padre	0	1	2	3	4	5	6	7	8	9
				↕	↕	↕	↕			
Madre	2	3	1	4	5	7	8	0	9	6
Hijo	0	1	2	4	5	7	8	3	6	9
Hija	2	7	1	3	4	5	6	0	9	8

Figura 4.2. Cruce PMX<sup>1</sup>.

Se realizan dos cortes al azar y se intercambian las posiciones de la ventana central. El resto de valores se toman del original salvo que ya esté colocado. En este caso se busca la coincidencia en el individuo.

<sup>1</sup> No se están representando valores en binario. Los números representan la posición de los genes.

#### 4.2.2 CRUCE OX

Padre	0	1	2	3	4	5	6	7	8	9
Madre	2	3	1	4	5	7	8	0	9	6
Hijo	2	3	6	4	5	7	8	9	0	1
Hija	1	7	8	3	4	5	6	0	9	2

Figura 4.3. Cruce OX<sup>1</sup>.

Se toma una ventana central al azar con dos cortes y se intercambian los genes. El resto de genes se rellena a partir de los valores de la madre en orden.

#### 4.2.3 CRUCE CX

Padre	0	1	2	3	4	5	6	7	8	9
Madre	2	3	1	4	5	7	8	0	9	6
Hijo	0	1	2	3	4	5	8	7	9	6
Hija	2	3	1	4	5	7	6	0	8	9

Figura 4.4. Cruce CX<sup>1</sup>.

Los elementos vienen de los padres en la misma posición.

#### 4.2.4 INVERSIÓN SIMPLE

0	1	2	3	4	5	6	7	8	9
0	1	2	6	5	4	3	7	8	9

Figura 4.5.. Inversión Simple<sup>1</sup>.

Consiste en tomar una ventana al azar del individuo e invertir las posiciones.

### 5 APLICACIÓN AL PROBLEMA DE LA MOCHILA

Un problema típico de programación es el que nos ocupa, “el problema de la mochila [4]”, que responde a la siguiente situación: imagínese hacer una excursión a la que solo podemos llevar una mochila que, lógicamente, tiene una capacidad limitada. Cada objeto que introducimos ocupa un volumen dentro de la misma y en contrapartida durante el viaje nos proporcionará un beneficio o utilidad (ejemplo: una cantimplora), el problema surge cuando debemos elegir qué objetos seleccionar para llevar en la mochila de forma que nuestro beneficio sea máximo (tengamos todo lo necesario) sin exceder su capacidad.

Los datos del problema son los siguientes:

- $n$ : número de objetos entre los que se puede elegir.
- $p_i$ : peso del objeto  $i$ -ésimo que no es más que una forma de hacer referencia a un objeto cualquiera que pueda ser incluido en la mochila -, es decir,  $c_i$  representa el coste de escoger un objeto, en tanto en cuanto va a ocupar un “espacio de la mochila” que dejará fuera otros objetos.
- $b_i$ : utilidad o beneficio que proporciona el objeto  $i$ -ésimo.
- $C$ : capacidad de la mochila, equivale al presupuesto máximo del que se dispone. Tomamos  $C = 40$ .

Los valores del ejemplo sencillo que vamos a tratar los recogemos en la Tabla 1.

Tabla 1. Tabla de pesos y beneficios.

$i$	1	2	3	4	5
$p_i$	30	20	22	10	7
$b_i$	100	75	80	40	20

Los elementos a introducir en la mochila van a ser nuestras variables objeto de análisis, cada variable la denotaremos como  $x_i$ . Se asignará el valor 1 si llevamos el objeto y con un 0 en el caso contrario.

La restricción vendrá marcada por la capacidad máxima de la mochila, de tal forma que la suma de todos los objetos multiplicados por el espacio que ocupan en la mochila no podrá exceder dicha capacidad máxima. Matemáticamente:

$$\sum_{i=1}^n p_i x_i \leq C. \quad (0.3)$$

Tal y como se expresa en la definición, el objetivo de este problema es seleccionar aquellos objetos que introducidos en la mochila nos proporcionan una mayor utilidad. En otras palabras, lo que debemos hacer es maximizar la utilidad de nuestra mochila.

Matemáticamente:

$$\max z = \sum_{i=1}^n b_i x_i. \quad (0.4)$$

Representamos a cada individuo con un cromosoma de 5 alelos, en el que el bit  $i$  hace referencia al objeto  $i$ . Por ejemplo, supongamos que tenemos el genotipo de la Figura 5.1: significa que nos llevaríamos los objetos 1, 3 y 5.

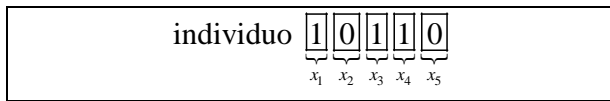


Figura 5.1. Ejemplo de genotipo.

La función de salud que vamos a utilizar en primera instancia es

$$z = \sum_{i=1}^n b_i x_i. \quad (0.5)$$

Si evaluamos el individuo anterior, tenemos que

$$z = \sum_{i=1}^n b_i x_i = 100 + 80 + 1 = 220 \quad (0.6)$$

Comprobamos la condición de capacidad:

$$\sum_{i=1}^n p_i x_i = 30 + 22 + 10 = 62 > 40 \quad (0.7)$$

Este individuo incumple la restricción de capacidad de la mochila. Es un individuo no factible.

**Discutir la posibilidad de utilizar la función  $z$  para evaluar la salud de los individuos en la resolución del problema de la mochila 0-1 genérico mediante algoritmos genéticos. Determinar razonadamente una función apropiada para este propósito, en la que se tenga en cuenta el valor de  $z$ .**

No es adecuado porque el algoritmo evolucionaría a poner todo 1 y esa solución es NO Factible.

*¿Cómo evaluar los no factibles?* Podemos penalizarlos buscando una expresión matemática adecuada. La restricción

$\sum p_i \cdot x_i \leq C$  es de tipo lineal, por lo que buscaremos una penalización lineal, por ejemplo, cogemos lo que me paso de peso y lo multiplicamos por un factor  $k$ .

De esta manera, la función de salud queda:

$$s = \sum_{i=1}^n b_i x_i - k \cdot \left( \sum_{i=1}^n p_i x_i - C \right)^+, \quad (0.8)$$

donde,

$$\left( \sum_{i=1}^n p_i x_i - C \right)^+ = \begin{cases} 0 & \text{si } \sum_{i=1}^n p_i x_i \leq C \\ \sum_{i=1}^n p_i x_i - C & \text{si } \sum_{i=1}^n p_i x_i > C \end{cases} \quad (0.9)$$

*¿Cuánto vale  $k$ ?* El tamaño de  $k$  puede variar en función de  $n$ . También habría que estudiar la no-factibilidad en función de  $k$ : un  $k$  muy grande puede dar lugar a valores de salud negativos, lo cual es absurdo. Esto se podría arreglar sumando una cota, pero podríamos entrar en problemas de no-discriminación.

**Obtener una población inicial de 6 individuos por el método de la muestra invertida.**

El método consiste en generar la mitad de la población de forma aleatoria y la otra mitad serán los inversos de los primeros, tal y como se observa en la Figura 5.1.

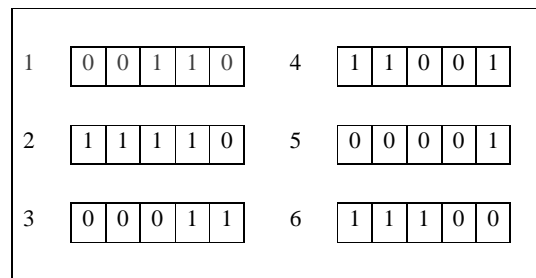


Figura 5.1. Muestra invertida.

**Seleccionar los padres de la siguiente generación por el método de la ruleta a partir de los generados anteriormente.**

Tomamos un valor de  $k = 7$ . Evaluamos los individuos anteriores y comprobamos la restricción de capacidad.

$$\boxed{1} \begin{cases} s = 80 + 40 = 120 \\ \text{peso} = 22 + 10 = 32 \end{cases}$$



$$\boxed{2} \begin{cases} s = 100 + 75 + 80 + 40 - 7 \cdot (82 - 40) = 1 \\ \text{peso} = 30 + 20 + 22 + 10 = 82 \Rightarrow \text{No Factible} \end{cases}$$

$$\boxed{3} \begin{cases} s = 40 + 20 = 60 \\ \text{peso} = 10 + 7 = 17 \end{cases}$$

$$\boxed{4} \begin{cases} s = 100 + 75 + 20 - 7 \cdot (57 - 40) = 76 \\ \text{peso} = 30 + 20 + 7 = 57 \Rightarrow \text{No Factible} \end{cases}$$

$$\boxed{5} \begin{cases} s = 20 \\ \text{peso} = 7 \end{cases}$$

$$\boxed{6} \begin{cases} s = 100 + 75 + 80 - 7 \cdot (72 - 40) = 31 \\ \text{peso} = 30 + 20 + 22 = 72 \Rightarrow \text{No Factible} \end{cases}$$

[4] Apuntes de la asignatura Métodos Cuantitativos de Organización II, Miguel Gutiérrez – Universidad Carlos III de Madrid

[5] Algoritmos Genéticos, Abdelmalik Moujahid, Iñaki Inza y Pedro Larrañaga - Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad del País Vasco.

**Tabla 2. Evaluación de los individuos.**

<i>Individuo</i>	<i>Salud</i>	<i>Salud Relativa</i>
1	120	0,3896
2	1	0,0032
3	60	0,1948
4	76	0,2468
5	20	0,0649
6	31	0,1006
Total	308	1

Al hacer la ruleta entran individuos No Factibles a ser padres, por lo que no es un buen método de selección. Por lo tanto, diseñamos el método de selección “matando” a los individuos no factibles para que no sean padres de la siguiente generación. De esta manera, sólo los individuos 1, 3 y 5 pueden ser padres. Por el método de la ruleta se procedería a asignar cuántos hijos tendría cada individuo, hasta completar una población de 6 individuos.

## 6 REFERENCIAS

- [1] Apuntes asignatura Inteligencia Artificial en Redes de comunicación, Julio Villena Román – Universidad Carlos III de Madrid  
<http://www.it.uc3m.es/jvillena/irc/>
- [2] Algoritmos genéticos – Wikipedia, Enciclopedia libre  
[http://es.wikipedia.org/wiki/Algoritmo\\_genético](http://es.wikipedia.org/wiki/Algoritmo_genético)
- [3] Tutorial - Introduction to Genetic Algorithms  
<http://www.obitko.com/tutorials/genetic-algorithms/>