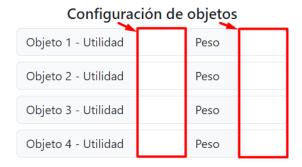
# **MANUAL DE USUARIO**

# **EJEMPLO DE LA MOCHILA**

### **EJECUCIÓN DEL ALGORTIMO**

 Como primera medida, debemos ingresar los valores referidos a la utilidad y peso de cada uno de los objetos. En cuanto a los pesos, tener que en cuenta que no deben superar el valor correspondiente a la capacidad máxima de la mochila ingresado en el siguiente paso.



2) Luego vamos a introducir los valores necesarios para la simulación referidos al peso máximo de la mochila, probabilidad de cruce, probabilidad de mutación y cantidad de iteraciones; todos los parámetros mencionados deben cumplir con las restricciones que se encuentran en pantalla.

### Configuración de simulación

Peso máximo de la mochila	1 ≤ x ≤ 100
Probabilidad de cruce	0 ≤ x ≤ 1
Probabilidad de mutación	0 ≤ x ≤ 1
Cantidad de iteraciones	1 ≤ x ≤ 100



3) A continuación, debemos seleccionar los bits correspondientes a los individuos de manera manual utilizando el menú desplegable " " o bien haciendo click en la opción "Generar bits aleatorios". Tanto los bits ingresados de manera manual como los generados de forma aleatoria deberán cumplir con la restricción de no superar el valor referido al peso máximo de la mochila ingresado en el paso anterior, ya que de ser así se informará al respecto mediante un mensaje de error.



4) Finalmente hacemos click en "Ejecutar algoritmo" para iniciar la simulación. Mediante la opción "Limpiar campos", tenemos la posibilidad de borrar todos los valores introducidos tanto binarios como decimales.

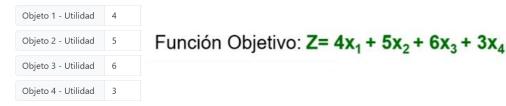


# INTERPRETACIÓN DE LOS RESULTADOS

1) Mediante las utilidades de los objetos ingresados, definimos la función objetivo a los fines de maximizar el espacio en la mochila, tal y como se muestra a continuación,

$$\mathsf{Maximizar} \quad \sum_{i=1}^n U_i x_i$$

A modo de ejemplo utilizaremos las siguientes utilidades para definir la función objetivo (también llamada función Fitness):



2) Además, debemos tener en cuenta las restricciones en cuanto al peso de cada uno de los objetos, como así también la capacidad total de la mochila:

Restricciones Sujeto a: 
$$\sum_{i=1}^{n} P_i x_i \leq \frac{Capacidad}{mochila}$$

En el siguiente ejemplo utilizaremos los siguientes pesos para definir las restricciones:



3) A continuación, debemos armar la tabla correspondiente a la primera iteración, ubicando los bits de cada individuo para calcular la utilidad y el peso total de los mismos como así también la probabilidad y su acumulada.

### Configuración de bits de individuos

Individuo 1	1 ~	1 ~	0 ~	0 ~
Individuo 2	0 ~	1 ~	1 ~	0 ~
Individuo 3	0 ~	1 ∨	0 ~	1 🗸
Individuo 4	1 ~	0 ~	1 ~	0 ~





### Ejemplo Individuo 1:

Utilidad total = 4x1+5x1+6x0+3x0 = 9

Peso total = 7x1+6x1+8x0+2x0 = 13

Probabilidad = 9/38 = 0.24

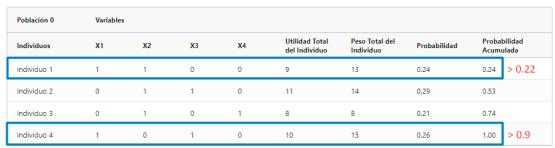
- 4) Luego de calcular las utilidades, pesos y probabilidades, se deben seleccionar los padres para realizar tantos cruces como sean necesarios y de esa manera obtener los cuatro hijos que integrarán la población para la iteración siguiente.
- 5) El simulador asigna dos probabilidades aleatorias (en color azul) para elegir dos individuos de la Tabla de población de padres.

# Cruce 1 Cruce 2 Cruce 3 Elección de Padres Probabilidad padre 1 0.9 Individuo elegido 4 Bits 1 0 1 0 Probabilidad padre 2 0.22 Individuo elegido 1 Bits 1 1 0 0



6) Para la elección de los individuos, los valores correspondientes a las probabilidades aleatorias son comparados con los de la columna probabilidad acumulada de la tabla Población de padre, y aquel valor acumulado que primero supere la probabilidad aleatoria es considerado padre. Tal y como se muestra en la figura.

### Población de Padres



### **Cruces entre Padres**



7) En cuanto al cruzamiento, el simulador asigna una probabilidad aleatoria (en color azul) de cruce y si la misma resulta ser menor que la probabilidad de cruce ingresada como dato al inicio de la simulación, entonces los bits de los padres seleccionados en el inciso anterior se cruzarán.

### Cruzamiento



8) A continuación, se asigna de manera aleatoria una probabilidad de punto de corte para definir entre cuáles genes se encuentra.

Entre gen 1 y gen 2 -> Probabilidad de 0 - 0.33

Entre gen 2 y gen 3 -> Probabilidad de 0.33 - 0.66

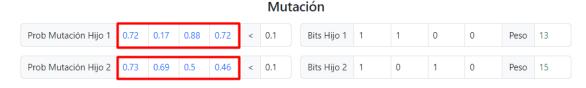
Entre gen 3 y gen 4 -> Probabilidad de 0.66 – 0.99



Tal y como se muestra en el siguiente ejemplo, el punto de corte se encuentra entre los genes 1 y 2.



9) Seguidamente, se encuentra el proceso de Mutación para el cual se asignará una probabilidad aleatoria de mutación para cada uno de los genes que integran a los hijos. Luego se verifica si cada probabilidad aleatoria resulta menor a la probabilidad de mutación ingresada al inicio de la simulación. En caso de ser menor el bit de ese gen debe ser reemplazado por el opuesto, de lo contrario la estructura del cromosoma no se modifica (Tal y como se muestra en el ejemplo).



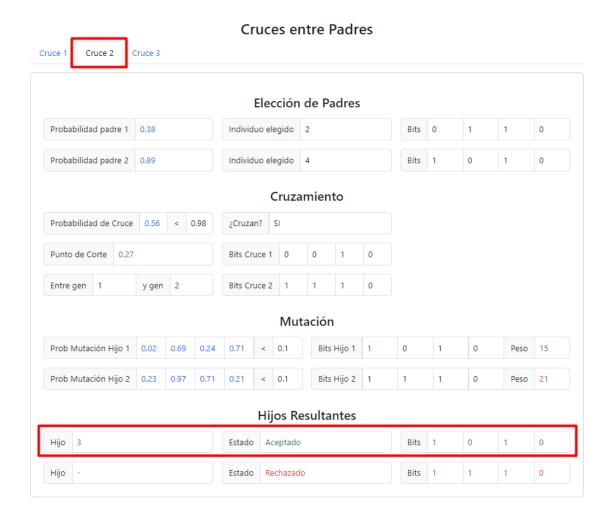
10) Finalmente, debemos calcular los pesos totales de cada uno de los posibles hijos utilizando la restricción establecida al inicio de la simulación, y de esa manera podremos definir si dichos hijos resultan aceptados o rechazados para integrar la nueva población. Ejemplo:

Peso hijo 1 = 7x1+6x1+8x0+2x0 = 13





El procedimiento de cruce de los padres que se mencionó anteriormente se repetirá en la misma iteración hasta hallar los cuatro hijos que formarán parte de la nueva población (como población de padres) utilizada en la siguiente iteración. A modo ilustrativo se muestra el cruce 2 (Donde se obtienen un hijo) y cruce 3 (Donde se obtiene el hijo restante), como así también la población final de la primera iteración:





Prob Mutación Hijo 2 0.81

Hijo

Hijo 4

0.73

0.55

# UNIVERSIDAD TECNOLÓGICA NACIONAL – FACULTAD REGIONAL CÓRDOBA DEPARTAMENTO DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN CÁTEDRA DE INTELIGENCIA ARTIFICIAL

### **Cruces entre Padres** Cruce 3 Cruce 1 Elección de Padres Probabilidad padre 1 0.72 Individuo elegido 3 Bits 0 Probabilidad padre 2 0.87 Individuo elegido 4 0 0 Bits 1 Cruzamiento ¿Cruzan? SI Probabilidad de Cruce 0.01 < 0.98 Punto de Corte 0.44 Bits Cruce 1 0 Entre gen 2 Bits Cruce 2 1 Mutación Prob Mutación Hijo 1 0.96 0.21 Bits Hijo 1 0 Peso

0.1

Rechazado

Aceptado

Estado

Estado

Hijos Resultantes

Luego de obtener los cuatro hijos con sus respectivos bits, a continuación, se muestra como quedaría la población de hijos resultante de la iteración N°1, la cual es definida como la población de Padres en la iteración N°2:

Bits Hijo 2 1

0

Bits 0

Bits 1

1

0

0

1

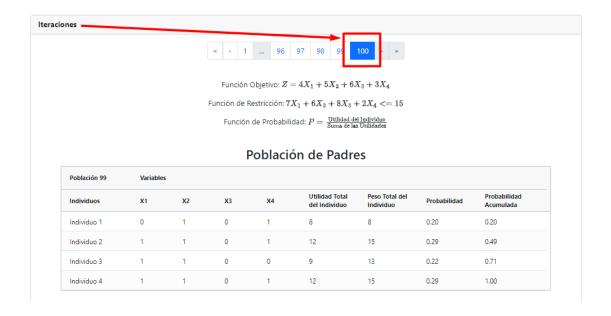
### Población de Hijos



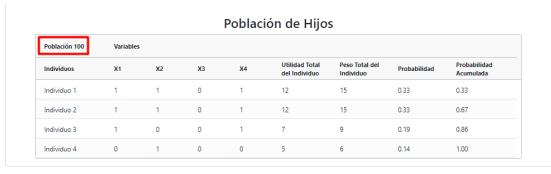




Una vez que se ejecutaron la totalidad de las iteraciones indicadas al principio de la simulación, obtenemos la población final resultante.







Población 100	Variable	Variables								
Individuos	Х1	X2	Х3	X4	Utilidad Total del Individuo	Peso Total del Individuo	Probabilidad	Probabilidad Acumulada		
Individuo 1	1	1	0	1	12	15	0.33	0.33		
Individuo 2	1	1	0	1	12	15	0.33	0.67		
Individuo 3	1	0	0	1	7	9	0.19	0.86		
Individuo 4	0	1	0	0	5	6	0.14	1.00		