

Manual de usuario

Programación Entera Pura

Introducción

Dentro de los métodos para la optimización y solución de problemas está la programación entera. Las variables, en este modelo de decisión, toman valores restringidos a números enteros, ya que por ejemplo en la optimización de una planta eléctrica no se pueden apagar y/o agregar un generador y medio; la programación entera se divide en dos, Entera mixta (PEM) y Entera Pura (PEP), esta última es en la que nos enfocaremos.

A través de este documento se realizará la conceptualización de la programación entera pura haciendo uso del contexto histórico, las metodologías de desarrollo y los software que implementan estos modelos. Asimismo se hará uso de ejemplos triviales y de aplicación para acoplar y afianzar este modelo matemático de optimización.

Requisitos

El Software para poder usarse se debe acceder a una página web en la que el código se ejecuta, por este método no se debe instalar nada, único requisito es tener conexión a internet ya que el software está en la nube y se ejecuta en los servidores de Google Colab

Si se desea ejecutar en el pc se debe tener python e instalar el "CVXOPT", para ser instalado se debe abrir el símbolo del sistema usando "pip" con el siguiente comando

```
pip install cvxopt
```

También se puede seguir los pasos que estan en la pagina de cvxopt

<https://cvxopt.org/install/index.html>

O se pueden descargar los archivos necesarios para su instalación

<https://cvxopt.org/download/index.html>

Uso de software

1. Ingresamos en la matriz “c” los valores asociados a la función objetivo, es decir los valores que acompañan a las variables $X_1, X_2, X_3, \dots, X_n$, estos valores separados por comas y dentro de unos corchetes, en este caso si la función objetivo es:

$$Z = 4X_1 + 3X_2 + 4X_3$$

se debe ingresar así

```
c = m([-4., -3., -4.]) #Matriz con los valores de la función objetivo
```

2. En la matriz “A” Ingresamos los valores de las variables de restricción que acompañan a las variables $X_1, X_2, X_3, \dots, X_n$, estas también separadas por comas, los valores de todas las X_i van encerradas en corchetes, si hay mas variables se separa por coma y se agrega otro corchete, por ejemplo si las restricciones fueran

$$5X_1 + 2X_2 + 1X_3 \leq 5$$

$$2X_1 + 4X_2 + 7X_3 \leq 10$$

Los valores se deben ingresar así:

```
A = m([[5., 2.], [2., 4.], [1., 7.]]) #Valores de las variables de las restricciones
```

en donde el primer corchete corresponde a X_1 , el segundo a X_2 y el tercero a X_3

3. En la matriz “b” Ingresamos los valores independientes de las restricciones es decir los que van después del \leq o \geq , estos valores están separados por comas dependiendo la cantidad de restricciones que hayan

Con las restricciones anteriores seria:

```
b = m([5., 10.]) #Valores independientes de las restricciones
```

4. Especificamos que las variables sean enteras y binarias, reemplazando la “x” por la cantidad de variables que hay, en este caso hay X_1, X_2 y X_3 entonces son 3 variables

```
intVars = range(3) #Especificamos que las x variables son enteras  
binVars = range(3) #Especificamos que las x variables son binarias
```

5. Damos correr y este ejecutará y mostrará el resultado, lo que sale en valores óptimos de las variables el resultado saldrá dándonos 1 o 0, el primero que sale corresponde a X1 y así sucesivamente, el 1 significa que debe usarse esa variable y su valor asociado si es 0 significa que no debe usarse, al final nos dará el valor óptimo que debe dar

```
Los valores óptimos de las variables son:
```

```
[ 0.00e+00]
```

```
[ 0.00e+00]
```

```
[ 1.00e+00]
```

```
El valor óptimo es: 4.0
```

En este caso el resultado nos dice que se debe tomar la variable X3 y que el valor óptimo es $Z = 4$