|  |
| --- |
| **PROGRAMACIÓN ENTERA BINARIA**  Luis Felipe Corredor Espinosa – 20171020056,  Cristian David Rodríguez – 20171020072.  Noviembre 2020  Universidad Distrital Francisco José de Caldas  Facultad de Ingeniería – Ingeniería de Sistemas  **Link del repositorio:**  <https://github.com/cristianrodriguez05/Investigacion-de-operaciones-1> **Resumen** En el presente documento se aborda la aplicación de la programación entera binaria para la solución de problemas, sus diversas implicaciones, se realiza la presentación de algunos ejercicios solucionados, así como también el desarrollo del algoritmo de la programación entera binaria en código Python con interfaz de usuario que nos permita la solución práctica de cualquier problema correspondiente a variables enteras binarias.  **Tabla de contenidos**  [**Resumen** 2](#_Toc59020285)  [**Introducción** 4](#_Toc59020286)  [**Objetivos** 4](#_Toc59020287)  [**Historia** 6](#_Toc59020288)  [**Programación entera binaria** 7](#_Toc59020289)  [**Usos** 8](#_Toc59020290)  [**Ejemplos** 9](#_Toc59020291)  [**Problema de la Mochila** 9](#_Toc59020292)  [**Conclusiones** 11](#_Toc59020293)  [**Lista de referencias** 12](#_Toc59020294) |

|  |
| --- |
| **Introducción** En muchos problemas prácticos, las variables de decisión sólo tienen sentido real si su valor es entero. Por ejemplo, con frecuencia es necesario asignar a las actividades cantidades enteras de personas, máquinas o vehículos. Si el hecho de exigir valores enteros es la única diferencia que tiene un problema con la formulación de programación lineal, entonces se trata de un problema de programación entera **(PE)**.  La programación entera binaria es un método perteneciente a la programación lineal, por lo que su base es un algoritmo matemático que tiene como finalidad resolver un problema indeterminado formulado a través de ecuaciones lineales, optimizando así una función objetivo también lineal que generalmente se refiere a costo o a tiempo [[1].](#_Lista_de_referencias) **Objetivos**  * Entender el concepto y condiciones de un problema de programación entera lineal binaria. * Conocer y aplicar el procedimiento de solución de problemas correspondientes a la programación entera lineal binaria. * Implementación de programación entera lineal binaria a código con interfaz de usuario.  **Historia** El problema de resolver un sistema de desigualdades lineales se remonta al menos hasta Fourier , quien en 1827 publicó un método para resolverlas, y que da nombre al método de eliminación de Fourier-Motzkin .  En 1939, el matemático y economista soviético Leonid Kantorovich , quien también propuso un método para resolverlo, dio una formulación de programación lineal de un problema que es equivalente al problema general de programación lineal . Dantzig desarrolló de forma independiente una formulación de programación lineal general para utilizarla en problemas de planificación en la Fuerza Aérea de EE. En 1947, Dantzig también inventó el método simplex que por primera vez abordó de manera eficiente el problema de la programación lineal en la mayoría de los casos . Sin embargo, solo lleva un momento encontrar la solución óptima planteando el problema como un programa lineal y aplicando el algoritmo simplex .  La teoría detrás de la programación lineal reduce drásticamente el número de posibles soluciones que deben verificarse. Leonid Khachiyan demostró por primera vez que el problema de programación lineal se podía resolver en tiempo polinómico en 1979, pero un gran avance teórico y práctico en el campo se produjo en 1984 cuando Narendra Karmarkar introdujo un nuevo método de punto interior para resolver la programación lineal.  Para la Programación Lineal Entera Binaria el método de solución más utilizado es el Método de Ramificación y Acotamiento. |

# **Programación entera binaria**

La programación de enteros **0-1** o la programación de enteros binarios **(BIP)** es el caso especial de la programación de enteros donde se requiere que las variables sean 0 o 1 (en lugar de enteros arbitrarios). [[2]](#_Lista_de_referencias)

Supongamos ahora que las variables pueden tomar solo valores 0 o 1.

Ahora no se tienen que agregar restricciones adicionales a los subproblemas sino determinar que una variable toma valor 0 o toma valor 1.

El número máximo de problemas a examinar es **O(2n)**, para un problema con n

variables binarias **(profundidad n)**.

La relajación lineal se obtiene reemplazando cada restricción de binariedad por

**0 ≤ xj ≤ 1** para cada variable.

Veamos un ejemplo:

Sea **(P0)** la relajación lineal.

La solución óptima viene dada por:

Ratificamos en **x3 (única posibilidad)**:

En **(P1)**, agregamos **x3 = 0**.

En **(P2)**, agregamos **x3 = 1**.

Luego de resolver el problema por **B&B** el árbol queda:

|  |
| --- |
| P0 |
| x1 = 1  x2 = 0  x3 = 1/2  z0 = 3 |

x3 = 1

x3 = 0

|  |
| --- |
| P2 |
| x1 = 1/2  x2 = 0  x3 = 1  z2 = 3 |

|  |
| --- |
| P1 |
| x1 = 1  x2 = 1/2  x3 = 0  z1 = 2.5 |

x2 = 0

|  |
| --- |
| P3 |
| x1 = 0  x2 = 1/2  x3 = 1  z3 = 2.5 |

|  |
| --- |
| P4 |
| x2 = 1 |

|  |
| --- |
| P7 |
| x1 = 1  x2 = 0  x3 = 0  z7 = 2 |

|  |
| --- |
| P8 |
| x1 = 1/2  x2 = 1  x3 = 0  z8 = 2 |

x1 = 0

x1 = 1

x2 = 1

x2 = 0

|  |
| --- |
| P5 |
| x1 = 0  x2 = 0  x3 = 1  z5 = 2 |

|  |
| --- |
| P6 |
|  |

# *Ilustracion1: Árbol resultante por método BB*

# **Usos**

La programación lineal es un campo de optimización ampliamente utilizado por varias razones. Varios algoritmos para otros tipos de problemas de optimización funcionan resolviendo problemas de LP como subproblemas. Asimismo, la programación lineal se utilizó mucho en la formación inicial de la microeconomía y actualmente se utiliza en la gestión empresarial, como la planificación, la producción, el transporte, la tecnología y otros temas.

# **Ejemplos**

## **Problema de la Mochila**

El problema de la mochila es uno de los 21 problemas NP-completos de Richard Karp, establecidos en un famoso artículo de 1972. [[3]](#_Lista_de_referencias)

Se tienen n tipos diferentes de objetos, cada uno de ellos tiene un peso **wj** y un valor

**vj**.

Se dispone de una mochila donde se colocarán los objetos, que soporta un peso máximo

de **W**, de manera de maximizar el valor total del contenido de la mochila.

Los objetos son indivisibles, o sea solo se puede colocar un número entero de cada

objeto.

Modelo: (Problema de la Mochila (knapsack) Entero)

xj → unidades del objeto **j** que se colocaran en la mochila.

Si solo existe un objeto de cada tipo entonces xj = 0 o 1 → knapsack binario o 0-1,

1 corresponde a poner el objeto en la mochila y 0 a no ponerlo.

Otra aplicación de este problema se utiliza cuando existen mercaderías que deben

ser almacenadas o transportadas considerando una disponibilidad de espacio o peso

limitado, con cada mercadería con un cierto valor.

Está probado que se puede resolver (usando programación dinámica) en tiempo

**O(nW)**. Pero si W no está acotado por un polinomio en n esto no es un buen

resultado.

# **Conclusiones**

* La programación entera binaria es adecuada para la resolución de problemas de toma de decisiones, Porque permite optimizar la decisión al comparar las diferentes alternativas.
* Es adecuada la programación entera binaria en problemas prácticos de la vida cotidiana done se tenga que tomar decisión a razón de limitación en espacio, peso o alguna clase de magnitud específica, permitiendo evaluar la eficiencia de las combinaciones.
* Es aprovechable en la optimización de procesos a nivel industrial, al hacer previamente un análisis de requerimientos, permitiendo optar por las mejores alternativas de acuerdo a las necesidades.

# **Lista de referencias**

• [[1]](#_Introducción) Hamdy A. Taha. “INVESTIGACIÓN DE OPERACIONES”. Novena edición.University of Arkansas, Fayetteville.

• [[2]](#_Programación_entera_binaria) ESCOBAR ALVARÁN Daniel Felipe, GARCÉS HINCAPIÉ Julián Alberto, RESTREPO CORREA Jorge Hernán. (2012). “Aplicación de la programación entera binaria para resolver el problema simple de balanceo de línea de ensamble: un caso de estudio”. Universidad Tecnológica de Pereira. <https://www.redalyc.org/pdf/849/84923878013.pdf>

• [[3]](#_Problema_de_la) KARP Richard M. (1972). “Reducibility Among Combinatorial Problems” .ed. Complexity of Computer Computations. New York: Plenum. pp. 85-103.