\_ **\_\_\_**

**( \_ )**

**/ \_ \**

**| (\_) |**

**\\_\_\_/**

**\_\_\_\_ \_ \_\_**

**| \_ \ \_ \_\_ \_\_\_ \_\_ \_ \_ \_\_ \_\_ \_ \_ \_\_ \_\_\_ \_\_ \_ \_\_\_(\_) /\_/ \_ \_\_**

**| |\_) | '\_\_/ \_ \ / \_` | '\_\_/ \_` | '\_ ` \_ \ / \_` |/ \_\_| |/ \_ \| '\_ \**

**| \_\_/| | | (\_) | (\_| | | | (\_| | | | | | | (\_| | (\_\_| | (\_) | | | |**

**|\_| |\_| \\_\_\_/ \\_\_, |\_| \\_\_,\_|\_| |\_| |\_|\\_\_,\_|\\_\_\_|\_|\\_\_\_/|\_| |\_|**

**|\_\_\_/**

**\_\_\_\_\_ \_**

**| \_\_\_\_|\_ \_\_ | |\_ \_\_\_ \_ \_\_ \_\_ \_**

**| \_| | '\_ \| \_\_/ \_ \ '\_\_/ \_` |**

**| |\_\_\_| | | | || \_\_/ | | (\_| |**

**|\_\_\_\_\_|\_| |\_|\\_\_\\_\_\_|\_| \\_\_,\_|**

**\_\_\_\_ \_**

**\_\_\_ \_\_\_ \_ \_\_ / \_\_\_|\_\_\_ \_ \_\_| |\_ \_\_\_ \_\_\_**

**/ \_\_/ \_ \| '\_ \ | | / \_ \| '\_\_| \_\_/ \_ \/ \_\_|**

**| (\_| (\_) | | | | | |\_\_| (\_) | | | || \_\_/\\_\_ \**

**\\_\_\_\\_\_\_/|\_| |\_| \\_\_\_\_\\_\_\_/|\_| \\_\_\\_\_\_||\_\_\_/**

**Contenido**

**=========**

**1. Introducción.**

**2. Modelo Programación Lineal Entera.**

**3. El Problema de Manera Gráfica.**

**4. El Problema de la Mochila.**

**5. Un Problema Binario.**

**6. El Algoritmo de Gomory.**

**7. Primer Ejemplo de Gomory.**

**8. Segundo Ejemplo de Gomory.**

**9. Tercer Ejemplo con Representación Real.**

**10. Algunos Cuidados al Utilizar Cortes.**

**11. Resumen.**

**12. Ejercicios.**

**1. Introducción.**

**================**

**En el enunciado general de los problemas de programación lineal se mencionó el supuesto de divisibilidad. Esto significa que las variables de decisión pueden tomar valores fraccionales o reales. Sin embargo muy a menudo las variables de decisión deben ser enteras. Cuando se producen camiones, automóviles, licuadoras no se pueden producir partes fraccionarias.**

**En las decisiones anteriores se puede utilizar los sistemas lineales continuos y redondear los resultados para obtener las soluciones enteras. Sin embargo, el redondear puede producir soluciones por debajo o por encima del valor óptimo. Para evitar estos problemas se ha desarrollado un modelo de optimización diferente el cual se conoce como programación entera.**

**2. Modelo Programación Lineal Entera.**

**=====================================**

**Un modelo es de programación lineal entera si se puede escribir de la siguiente forma:**

**Maximizar: c1 x1 + c2 x2 + ... + cn xn = z**

**Con las**

**restricciones: a11 x1 + a12 x2 + ... + a1n xn <= b1**

**a21 x1 + a22 x2 + ... + a2n xn <= b2**

**. . .**

**. . .**

**. . .**

**am1 x1 + am2 x2 + ... + amn xn <= bm**

**con x1,x2,...,xn >= 0**

**y x1,x2,...,xp p<=n son valores enteros.**

**Algunas variantes de este modelo son:**

**() El modelo puede ser para maximizar o minimizar.**

**() Cada una de las restricciones puede relacionarse con bij mediante cualquiera de los símbolos = , <= , >=.**

**() Si todas las variables están restringidas a tomar valores enteros se dice que se trata de un problema de programación entera puro.**

**() Al contrario, si algunas variables deben tomar valores enteros y otras pueden tomar valores continuos se denomina un problema de programación entera mixto.**

**() Cuando las variables deben tomar valores de 0 o 1 únicamente se denomina un problema de programación entera 0-1 o binario.**

**3. El Problema de Manera Gráfica.**

**=================================**

**Suponga que se tiene el siguiente problema.**

**Función Objetivo:**

**(0) max z = 3 x1 + 4 x2**

**Sujeto a:**

**(1) 2 x1 + 1 x2 <= 6**

**(2) 2 x1 + 3 x2 <= 9**

**Donde:**

**(3) x1,x2 >= 0 y enteros**

**De forma compacta se representará como:**

**(0) max z = 3 x1 + 4 x2**

**(1) 2 x1 + 1 x2 <= 6**

**(2) 2 x1 + 3 x2 <= 9**

**(3) x1,x2 >= 0 y enteros**

**A resolver este problema se encuentra una solución de punto flotante. La solución del problema de programación lineal se encuentra en el punto:**

**x1 = 2.25**

**x2 = 1.50**

**z = 12.75**

**Parecería correcto redondear si se busca una solución entera.**

**El valor de x1 se debe redondear de 2.25 hacia 2.00.**

**El valor de x2 si se redondea hacia arriba sale fuera de la región factible.**

**En general para preservar la factibilidad de la solución se debe redondear hacia abajo. Por esta razón el valor de "x2" debe redondearse hacia abajo, y la solución cae en el punto:**

**x1 = 2.00**

**x2 = 1.00**

**z = 10.00**

**Sin embargo ninguno de los dos casos anteriores son el valor óptimo. El punto óptimo de esta función se encuentra en:**

**x1 = 0.00**

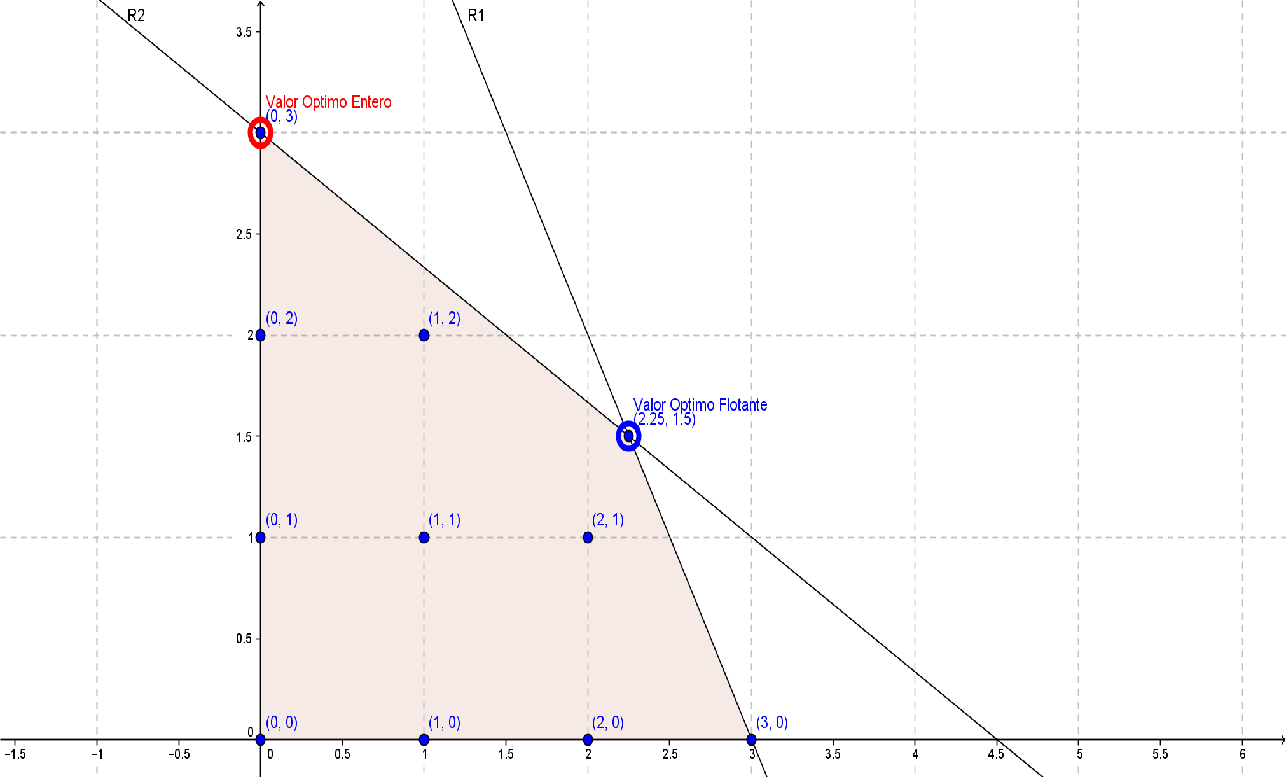
**x2 = 3.00**

**z = 12.00**

**El algoritmo de Simplex no puede encontrar esta solución por lo que es evidente que se necesita un nuevo tipo de algoritmo. A continuación se muestra un gráfico donde se muestra la situación explicada anteriormente.**

**### Gráfico.**

**### Solución Punto Flotante y Solución Entera.**

****

**4. El Problema de la Mochila.**

**=============================**

**Una persona se encuentra en una cueva donde hay tres tipos de piedras preciosas. Diamantes, esmeraldas y rubíes. Cada una de estas piedras tiene un cierto peso y valor, los cuales se describen en la siguiente tabla.**

**-------------------------**

**Piedra Peso Valor**

**kilos dólares**

**-------------------------**

**Diamante 4 1000**

**Esmeralda 3 700**

**Rubí 5 1200**

**-------------------------**

**La persona tiene un bolsa en la que puede llevar un máximo de 11 kilos. El problema consiste en decidir cuántas piedras de cada clase debe llevar de manera tal que no se exceda la capacidad de la bolsa. Las piedras no se pueden romper, pues son sumamente duras y no se cuenta con herramientas para cortarlas. Se desea maximizar la ganancia al momento de vender el contenido de la bolsa.**

**R/**

**Se inicia con la definición de las variables.**

**x1: cantidad de diamantes que se llevarán.**

**x2: cantidad de esmeraldas que se llevarán.**

**x3: cantidad de rubíes que se llevarán.**

**La función objetivo consistirá en maximizar el valor total que se lleva en la bolsa.**

**max z = 1000 x1 + 700 x2 + 1200 x3**

**El peso total de las joyas no puede exceder el límite de la mochila.**

**4 x1 + 3 x2 + 5 x3 <= 11**

**Finalmente no se pueden romper las piedras, por lo que las variables deben tomar valores enteros.**

**x1,x2,x3 >= 0 y enteros**

**Al unir los elementos anteriores se tiene el siguiente problema:**

**Función Objetivo:**

**(0) max z = 1000 x1 + 700 x2 + 1200 x3**

**Sujeto a:**

**(1) 4 x1 + 3 x2 + 5 x3 <= 11**

**Con las variables:**

**(2) x1,x2,x3 >= 0 y enteros**

**El anterior problema se puede escribir de forma resumida como:**

**(0) max z = 1000 x1 + 700 x2 + 1200 x3**

**(1) 4 x1 + 3 x2 + 5 x3 <= 11**

**(2) x1,x2,x3 >= 0 y enteros**

**5. Un Problema Binario.**

**=======================**

**Una compañía de desarrollo de software desea realizar hasta cinco proyectos. Sin embargo debido a las limitaciones de capital no puede seleccionar todos los proyectos que desea. En la actualidad cuenta con $60 para invertir en proyectos. A continuación se muestra una tabla con el costo de desarrollo de cada proyecto así como la ganacia esperada si se realiza.**

**-------------------------**

**Proyecto Costo Ganancia**

**-------------------------**

**1 52 100**

**2 23 60**

**3 35 70**

**4 15 15**

**5 7 15**

**-------------------------**

**Se desean escoger los proyectos de manera tal que se maximicen las ganancias, sin violar las restricciones.**

**R/**

**Se utilizarán las siguientes variables de decisión.**

**x1: 1 si se realiza el proyecto 1, 0 en otro caso.**

**x2: 1 si se realiza el proyecto 2, 0 en otro caso.**

**x3: 1 si se realiza el proyecto 3, 0 en otro caso.**

**x4: 1 si se realiza el proyecto 4, 0 en otro caso.**

**x5: 1 si se realiza el proyecto 5, 0 en otro caso.**

**Se desean maximizar las ganancias por lo que la función objetivo será:**

**max z = 100 x1 + 60 x2 + 70 x3 + 15 x4 + 15 x5**

**Los proyectos que se desarrollen no pueden exceder el dinero que tiene la empresa para invertir.**

**52 x1 + 23 x2 + 35 x3 + 15 x4 + 7 x5 <= 60**

**Finalmente las variables son enteras, pero solamente pueden tomar los valores de 0 y 1. Para ello se pueden agregar estas restricciones:**

**x1 <= 1**

**x2 <= 1**

**x3 <= 1**

**x4 <= 1**

**x5 <= 1**

**De esta forma el problema se escribiría como:**

**(0) max z = 100 x1 + 60 x2 + 70 x3 + 15 x4 + 15 x5**

**(1) 52 x1 + 23 x2 + 35 x3 + 15 x4 + 7 x5 <= 60**

**(2) x1 <= 1**

**(3) x2 <= 1**

**(4) x3 <= 1**

**(5) x4 <= 1**

**(6) x5 <= 1**

**(7) x1,x2,x3,x4,x5 >=0 y enteras**

**Usualmente, los problemas que tienen la estructura anterior, suelen escribirse de manera abreviada de la siguiente forma:**

**(0) max z = 100 x1 + 60 x2 + 70 x3 + 15 x4 + 15 x5**

**(1) 52 x1 + 23 x2 + 35 x3 + 15 x4 + 7 x5 <= 60**

**(2) 0 <= x1,x2,x3,x4,x4 <= 1 y enteras**

**6. El Algoritmo de Gomory.**

**==========================**

**Históricamente uno de los primeros algoritmos para resolver problemas enteros era el método de "cortar planos". El método se usa para resolver modelos de programación entera puros. Un problema de programación lineal entera se llama "puro" cuando todas sus variables básicas, incluyendo las variables de holgura, deben ser enteras. De otra forma se denomina un problema lineal "mixto".**

**El algoritmo consiste en resolver el problema con el simplex. Si el resultado es entero se utilizan estas soluciones. En caso contrario se agregan unas restricciones adicionales. Estas restricciones se agregan para reducir o cortar el espacio de soluciones, de manera tal que la solución fraccionaria encontrada quede fuera de la región factible y así obligar al algoritmo a buscar una solución entera. El proceso termina tan pronto como se encuentra una solución entera. A continuación se muestran los detalles del algoritmo.**

**() Algoritmo.**

**-------------**

**(1) Utilice el método simplex para encontrar una solución óptima del problema, sin tomar en cuenta que es un problema entero.**

**(2) Examine la solución óptima. Deténgase si todas las variables básicas tienen valores enteros. De lo contrario, construya una nueva restricción de la fila (k) que tenga la parte fraccionaria (fk) más grande y agréguela al conjunto inicial de restricciones. La nueva restricción se construye con las partes fraccionarias positivas de la restricción de la siguiente manera:**

**\_\_**

**\**

**/\_ (fj xj) >= fk**

**\_\_**

**\**

**/\_ (fj xj) - sk + ak = fk**

**(3) Encuentre una nueva solución al problema, si la solución encontrada es entera deténgase, de lo contrario vuelva al paso 2.**

**() Cómo se construyen los cortes.**

**Si bien las fórmulas anteriores describen como se deben construir los cortes es importante señalar algunos detalles adicionales. La fórmula resumida para realizar los cortes es la siguiente.**

**Cambie cada coeficiente de la restricción escogida, incluyendo el lado derecho de la siguiente manera:**

**Regla 1: Escriba todos los números como un entero con su parte fraccionaria, la parte fraccionaria debe ser menor que 1.**

**Regla 2: Si hay un entero positivo o negativo, la fracción es 0 y el corte lleva valor 0.**

**Regla 3: Si hay una fracción positiva, el corte lleva el mismo valor. Ejemplo 1/3 en la restricción, 1/3 en el corte.**

**Regla 4: Si hay una fracción negativa, el corte lleva el complemento**

**positivo. Ejemplo -1/3 en la restricción, 2/3 en el corte.**

**Observe la restricción y el corte que genera:**

**(R) 3 x1 + 17/5 x2 - 2/5 x3 = 35/4**

**(R) 3 x1 + 3 2/5 x2 - 2/5 x3 = 8 3/4**

**(C) 2/5 x2 + 3/5 x3 >= 3/4**

**Observe la restricción y el corte que genera:**

**(R) -13/4 x1 + 2/5 x2 - 2/5 x3 = 47/6**

**(R) -3 1/4 x1 + 2/5 x2 - 2/5 x3 = 7 5/6**

**(C) 3/4 x1 + 2/5 x2 + 3/5 x3 >= 5/6**

**Observe la restricción y el corte que genera:**

**(R) 0 x1 - 3/4 x2 + 1/3 x3 - 13/4 x4 = 73/7**

**(R) 0 x1 - 3/4 x2 + 1/3 x3 - 3 1/4 x4 = 10 3/7**

**(C) 1/4 x2 + 1/3 x3 + 3/4 x4 >= 3/7**

**De igual forma se pueden hacer cortes con números reales:**

**(R) 2 x1 + 1.40 x2 - 1.25 x3 + 0.25 x4 = 3.75**

**(C) 0.40 x2 + 0.75 x3 + 0.25 x4 >= 0.75**

**7. Primer Ejemplo de Gomory.**

**============================**

**Se tiene el siguiente problema.**

**(0) max z = 1 x1 + 2 x2**

**(1) 2 x2 <= 7**

**(2) 1 x1 + 1 x2 <= 7**

**(3) 2 x1 <= 11**

**(4) x1,x2 >= 0 y enteros**

**Al introducir las variables de holgura el problema se convierte en:**

**(0) max z = 1 x1 + 2 x2**

**(1) 2 x2 + s3 = 7**

**(2) 1 x1 + 1 x2 + s4 = 7**

**(3) 2 x1 + s5 = 11**

**La primera tabla se construye como:**

**(0) max z - 1 x1 - 2 x2 = 0**

**(1) 2 x2 + s3 = 7**

**(2) 1 x1 + 1 x2 + s4 = 7**

**(3) 2 x1 + s5 = 11**

**>>> Iteración 0:**

**------------------------------------------**

**i BVS x1 x2 s3 s4 s5 RHS**

**------------------------------------------**

**0 z -1 -2 0 0 0 0**

**1 s3 0 2 1 0 0 7**

**2 s4 1 1 0 1 0 7**

**3 s5 2 0 0 0 1 11**

**------------------------------------------**

**Al resolver el problema de la manera usual con el simplex se obtiene la siguiente solución final.**

**>>> Tabla Final, utilizando simplex:**

**------------------------------------------**

**i BVS x1 x2 s3 s4 s5 RHS**

**------------------------------------------**

**0 z 0 0 1/2 1 0 21/2 na**

**1 x2 0 1 1/2 0 0 7/2 --> 3 1/2**

**2 x1 1 0 -1/2 1 0 7/2 --> 3 1/2**

**3 s5 0 0 1 -2 1 4 --> 4**

**------------------------------------------**

**Esta solución no es aceptable pues las soluciones no son enteras. Se ha obtenido la solución:**

**x1 = 7/2 = 3.50**

**x2 = 7/2 = 3.50**

**z = 21/2 = 10.50**

**Por lo tanto se debe construir una nueva restricción. Para ello se descomponen los resultados en su parte entera sumada de su parte fraccionaria.**

**Se escoge la fila con la parte fraccional más grande. Como hay dos filas con el mismo valor se tomará de forma arbitraria la fila 1. Se tiene que la primera restricción se comporta como:**

**(1 ) 0 x1 + 1 x2 + 1/2 s3 + 0 s4 + 0 s5 = 7/2**

**(1 ) 0 x1 + 1 x2 + 1/2 s3 + 0 s4 + 0 s5 = 3 1/2**

**(4G) 1/2 s3 >= 1/2**

**() Reoptimizando desde el final.**

**--------------------------------**

**Para resolver el problema inicial se toma la última tabla y se agrega la nueva restricción de Gomory. La última tabla se muestra a continuación.**

**Tabla Final, utilizando simplex:**

**------------------------------------------**

**i BVS x1 x2 s3 s4 s5 RHS**

**------------------------------------------**

**0 z 0 0 1/2 1 0 21/2**

**1 x2 0 1 1/2 0 0 7/2**

**2 x1 1 0 -1/2 1 0 7/2**

**3 s5 0 0 1 -2 1 4**

**------------------------------------------**

**Como esta restricción tiene un >= es necesario una nueva variable de holgura y una variable artificial. Por lo tanto se obtiene la tabla. Se debe agregar una nueva restricción a la tabla:**

**(4G) 1/2 s3 >= 1/2**

**Como en la tabla se tienen únicamente igualdades se debe convertir a una igualdad agregando una nueva variable de sobrante y una variable artificial:**

**(4G) 1/2 s3 - s6 + a7 = 1/2**

**Como se ha agregado una nueva variable artificial se debe volver a realizar la primera fase del algoritmo:**

**(0') min w = a7**

**(0') max -w = -a7**

**(0') max -w + a7 = 0**

**Finalmente se deben agregar estas dos ecuaciones (0') y (4G) a la tabla.**

**>>> Iteración 0:**

**Tabla Final obtenida al agregar (0') y (4G)**

**----------------------------------------------------**

**i BVS x1 x2 s3 s4 s5 s6 a7 RHS**

**----------------------------------------------------**

**0' -w 0 0 0 0 0 0 1 0**

**0 z 0 0 1/2 1 0 0 0 21/2**

**1 x2 0 1 1/2 0 0 0 0 7/2**

**2 x1 1 0 -1/2 1 0 0 0 7/2**

**3 s5 0 0 1 -2 1 0 0 4**

**4G a7 0 0 1/2 0 0 -1 1 1/2**

**----------------------------------------------------**

**-1 f4G + f0' -> f0'**

**----------------------------------------------------**

**i BVS x1 x2 s3 s4 s5 s6 a7 RHS**

**----------------------------------------------------**

**0' -w 0 0 -1/2 0 0 1 0 -1/2**

**0 z 0 0 1/2 1 0 0 0 21/2**

**1 x2 0 1 1/2 0 0 0 0 7/2**

**2 x1 1 0 -1/2 1 0 0 0 7/2**

**3 s5 0 0 1 -2 1 0 0 4**

**4G a7 0 0 1/2 0 0 -1 1 1/2**

**----------------------------------------------------**

**>>> Iteración 1, 1era Fase.**

**----------------------------------------------------**

**i BVS x1 x2 s3\* s4 s5 s6 a7 RHS**

**----------------------------------------------------**

**0' -w 0 0 -1/2 0 0 1 0 -1/2 na**

**0 z 0 0 1/2 1 0 0 0 21/2 na**

**1 x2 0 1 1/2 0 0 0 0 7/2 7**

**2 x1 1 0 -1/2 1 0 0 0 7/2 +oo**

**3 s5 0 0 1 -2 1 0 0 4 4**

**4G a7\* 0 0 1/2 0 0 -1 1 1/2 1**

**----------------------------------------------------**

**----------------------------------------------------**

**i BVS x1 x2 s3 s4 s5 s6 a7 RHS**

**----------------------------------------------------**

**0' -w 0 0 0 0 0 0 1 0**

**0 z 0 0 0 1 0 1 -1 10**

**1 x2 0 1 0 0 0 1 -1 3**

**2 x1 1 0 0 1 0 -1 1 4**

**3 s5 0 0 0 -2 1 2 -2 3**

**4G s3 0 0 1 0 0 -2 2 1**

**----------------------------------------------------**

**>>> Iteración 2, 2da Fase**

**-----------------------------------------------**

**i BVS x1 x2 s3 s4 s5 s6 RHS**

**-----------------------------------------------**

**0 z 0 0 0 1 0 1 10**

**1 x2 0 1 0 0 0 1 3**

**2 x1 1 0 0 1 0 -1 4**

**3 s5 0 0 0 -2 1 2 3**

**4G s3 0 0 1 0 0 -2 1**

**-----------------------------------------------**

**En la segunda fase se ha llegado al valor óptimo:**

**x1 = 4**

**x2 = 3**

**z = 10**

**() Reoptimizando desde el origen.**

**---------------------------------**

**También se puede resolver el problema si se modifica el problema original.**

**Problema original:**

**(0) max z = 1 x1 + 2 x2**

**(1) 2 x2 <= 7**

**(2) 1 x1 + 1 x2 <= 7**

**(3) 2 x1 <= 11**

**(4) x1,x2 >= 0 y enteros**

**Que se reescribe como:**

**(0) max z = 1 x1 + 2 x2**

**(1) 2 x2 + 1 s3 = 7**

**(2) 1 x1 + 1 x2 + 1 s4 = 7**

**(3) 2 x1 + 1 s5 = 11**

**(4) x1,x2 >= 0 y enteros**

**Se obtiene la última tabla.**

**Tabla Final, utilizando simplex.**

**------------------------------------------**

**i BVS x1 x2 s3 s4 s5 RHS**

**------------------------------------------**

**0 z 0 0 1/2 1 0 21/2 na**

**1 x2 0 1 1/2 0 0 7/2 --> 3 1/2**

**2 x1 1 0 -1/2 1 0 7/2 --> 3 1/2**

**3 s5 0 0 1 -2 1 4 --> 4**

**------------------------------------------**

**Se selecciona la restricción (1) y se construye el nuevo corte:**

**(1 ) 0 x1 + 1 x2 + 1/2 s3 + 0 s4 + 0 s5 = 7/2**

**(1 ) 0 x1 + 1 x2 + 1/2 s3 + 0 s4 + 0 s5 = 3 1/2**

**(4G) 1/2 s3 >= 1/2**

**Para resolver el problema inicial se escribe con sus variables de holgura y se agrega la nueva restricción. La restricción debe escribirse como una igualdad, por lo que se necesita una nueva variable de excedente.**

**Problema con holguras:**

**(0 ) max z = 1 x1 + 2 x2**

**(1 ) 2 x2 + 1 s3 = 7**

**(2 ) 1 x1 + 1 x2 + 1 s4 = 7**

**(3 ) 2 x1 + 1 s5 = 11**

**(4G) 1/2 s3 - 1 s6 = 1/2**

**Al resolver este problema mediante simplex, se obtendrá la siguiente solución:**

**Iteracion Final:**

**-------------------------------------------**

**BVS x1 x2 s3 s4 s5 s6 RHS**

**-------------------------------------------**

**z 0 0 0 1 0 1 10**

**x2 0 1 0 0 0 1 3**

**x1 1 0 0 1 0 -1 4**

**s5 0 0 0 -2 1 2 3**

**s3 0 0 1 0 0 -2 1**

**--------------------------------------------**

**Se ha llegado a la solución óptima:**

**x1 = 4**

**x2 = 3**

**z = 10**

**Grafico del Problema**

**--------------------**

**Al aplicar el corte, se crea una nueva restricción que quita una parte de la solución factible sin excluir ninguna de las posibles soluciones enteras. Explicaremos de manera algebraica como se construye ese nuevo corte.**

**El corte de Gomory está dado por:**

**1/2 s3 >= 1/2**

**La variable s3 es la holgura de la primera restricción:**

**2 x2 <= 7**

**2 x2 + s3 = 7**

**Despejamos s3 y sustituimos su valor en el corte de Gomory:**

**2 x2 + s3 = 7**

**s3 = 7 - 2 x2**

**Se sustituye en el corte:**

**1/2 s3 >= 1/2**

**1/2 (7 - 2 x2) >= 1/2**

**7/2 - x2 >= 1/2**

**- x2 >= 1/2 - 7/2**

**- x2 >= -6/2**

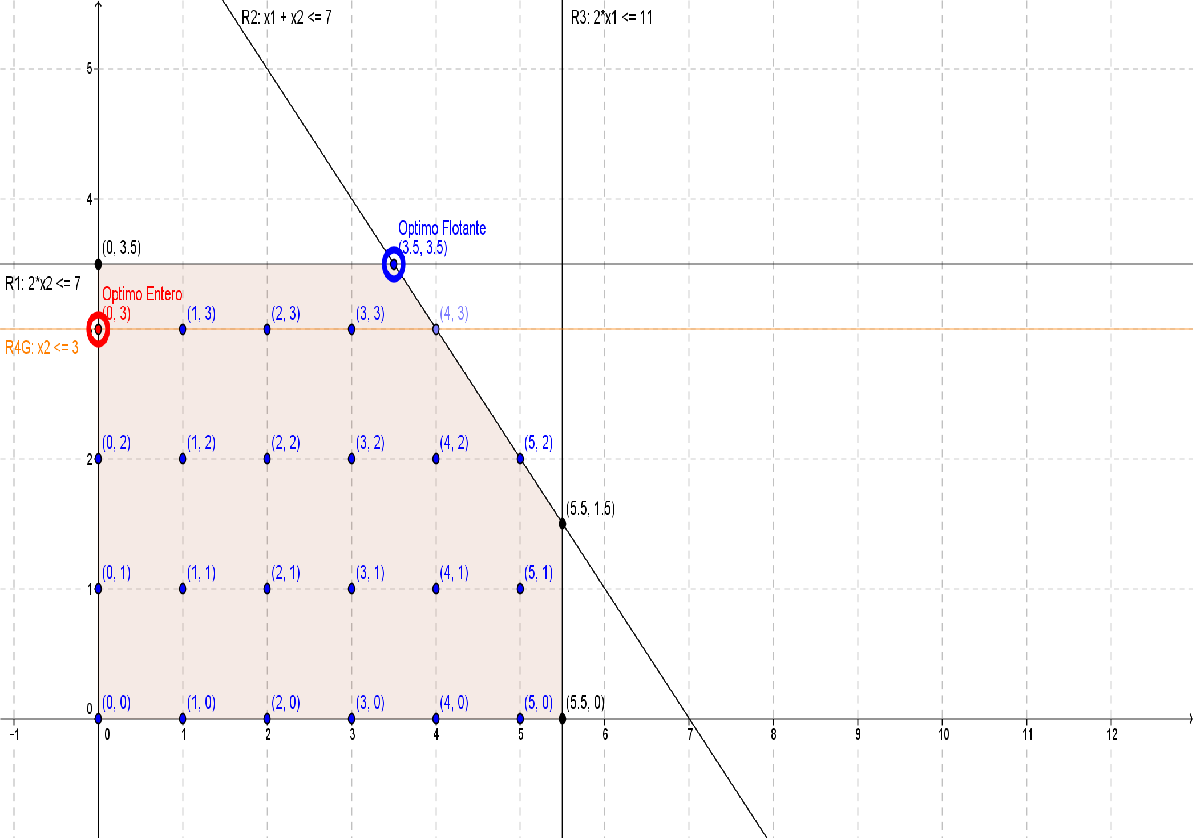
**- x2 >= -3**

**x2 <= 3**

**A continuación se presenta el gráfico original del problema y el respectivo corte de Gomory.**

**### Gráfico.**

**### Gráfico Original y Corte de Gomory.**

****

**8. Segundo Ejemplo de Gomory.**

**=============================**

**Se tiene el siguiente problema:**

**(0) max z = 1 x1 + 4 x2**

**(1) 2 x1 + 4 x2 <= 7**

**(2) 5 x1 + 3 x2 <= 15**

**(3) x1,x2 >= 0 y enteros**

**Se agregan las variables de holgura:**

**(0) max z = 1 x1 + 4 x2**

**(1) 2 x1 + 4 x2 + 1 s3 = 7**

**(2) 5 x1 + 3 x2 + 1 s4 = 15**

**(3) x1,x2 >= 0 y enteros**

**Al resolver el problema de la manera usual con el simplex se obtiene el siguiente resultado.**

**>>> Iteración 1, utilizando simplex:**

**----------------------------------------**

**i BVS x1 x2 s3 s4 RHS**

**----------------------------------------**

**0 z 1 0 1 0 7 na**

**1 x2 1/2 1 1/4 0 7/4 --> 1 3/4**

**2 s4 7/2 0 -3/4 1 39/4 --> 9 3/4**

**----------------------------------------**

**Se construye ahora la restricción de Gomory. Para ello se toma la fila con la parte fraccional más grande. En este caso se tomará la fila 1.**

**(1 ) 1/2 x1 + 1 x2 + 1/4 s3 + 0 s4 = 7/4**

**(1 ) 1/2 x1 + 1 x2 + 1/4 s3 + 0 s4 = 1 3/4**

**(3G) 1/2 x1 + 0 x2 + 1/4 s3 + 0 s4 >= 3/4**

**(3G) 1/2 x1 + 1/4 s3 >= 3/4**

**Problema modificado con el corte:**

**(0 ) max z = 1 x1 + 4 x2**

**(1 ) 2 x1 + 4 x2 + 1 s3 = 7**

**(2 ) 5 x1 + 3 x2 + 1 s4 = 15**

**(3G) 1/2 x1 + 1/4 s3 - 1 s5 = 3/4**

**Al resolver este problema mediante el simplex se obtiene la siguiente tabla:**

**>>> Iteración 2:**

**----------------------------------------------**

**i BVS x1 x2 s3 s4 s5 RHS**

**----------------------------------------------**

**0 z 0 0 1/2 0 2 11/2 na**

**1 x2 0 1 0 0 1 1 --> 1**

**2 s4 0 0 -5/2 1 7 9/2 --> 4 1/2**

**3G x1 1 0 1/2 0 2 3/2 --> 1 1/2**

**----------------------------------------------**

**Como la solución no es entera, se construye un nuevo corte de Gomory a partir de la fila 2:**

**(2 ) 0 x1 + 0 x2 - 5/2 s3 + 1 s4 + 7 s5 = 9/2**

**(2 ) 0 x1 + 0 x2 - 2 1/2 s3 + 1 s4 + 7 s5 = 4 1/2**

**(4G) 1/2 s3 >= 1/2**

**Se construye el nuevo problema:**

**(0 ) max z = 1 x1 + 4 x2**

**(1 ) 2 x1 + 4 x2 + 1 s3 = 7**

**(2 ) 5 x1 + 3 x2 + 1 s4 = 15**

**(3G) 1/2 x1 + 1/4 s3 - 1 s5 = 3/4**

**(4G) 1/2 s3 - 1 s6 = 1/2**

**Al resolver este problema se obtiene la siguiente tabla:**

**>>> Iteración 3:**

**----------------------------------------------------**

**i BVS x1 x2 s3 s4 s5 s6 RHS**

**----------------------------------------------------**

**0 z 0 0 0 0 2 1 5**

**1 x2 0 1 0 0 1 0 1**

**2 s4 0 0 0 1 7 -5 7**

**3G x1 1 0 0 0 -2 1 1**

**4G s3 0 0 1 0 0 -2 1**

**----------------------------------------------------**

**En este caso se tiene una solución entera, por lo que no se realizan más cortes.**

**x1 = 1**

**x2 = 1**

**z = 5**

**Gráfico del Problema**

**--------------------**

**Se puede apreciar el problema de forma gráfica. Para ello vamos a mostrar cuales son las nuevas restricciones que se deben agregar. Primero se debe despejar el valor de s3 dela R1.**

**Se toma R1:**

**2 x1 + 4 x2 <= 7**

**2 x1 + 4 x2 + s3 = 7**

**s3 = 7 - 2 x1 - 4 x2**

**Para el primer corte se utiliza 3G:**

**1/2 x1 + 1/4 s3 >= 3/4**

**1/2 x1 + 1/4 (7 - 2 x1 - 4 x2) >= 3/4**

**1/2 x1 + 7/4 - 2/4 x1 - 4/4 x2 >= 3/4**

**1/2 x1 + 7/4 - 1/2 x1 - 1 x2 >= 3/4**

**1/2 x1 - 1/2 x1 + 7/4 - 1 x2 >= 3/4**

**7/4 - 1 x2 >= 3/4**

**-1 x2 >= 3/4 - 7/4**

**-1 x2 >= -1**

**x2 <= 1**

**Para el segundo corte 4G se utiliza la restricción:**

**1/2 s3 >= 1/2**

**1/2 (7 - 2 x1 - 4 x2) >= 1/2**

**7/2 - 2/2 x1 - 4/2 x2 >= 1/2**

**7/2 - 1 x1 - 2 x2 >= 1/2**

**-1 x1 - 2 x2 >= 1/2 - 7/2**

**-1 x1 - 2 x2 >= -3**

**1 x1 + 2 x2 <= 3**

**Se puede observar en el gráfico siguiente como cada corte disminuye la región factible sin quitar ninguno de los puntos enteros:**

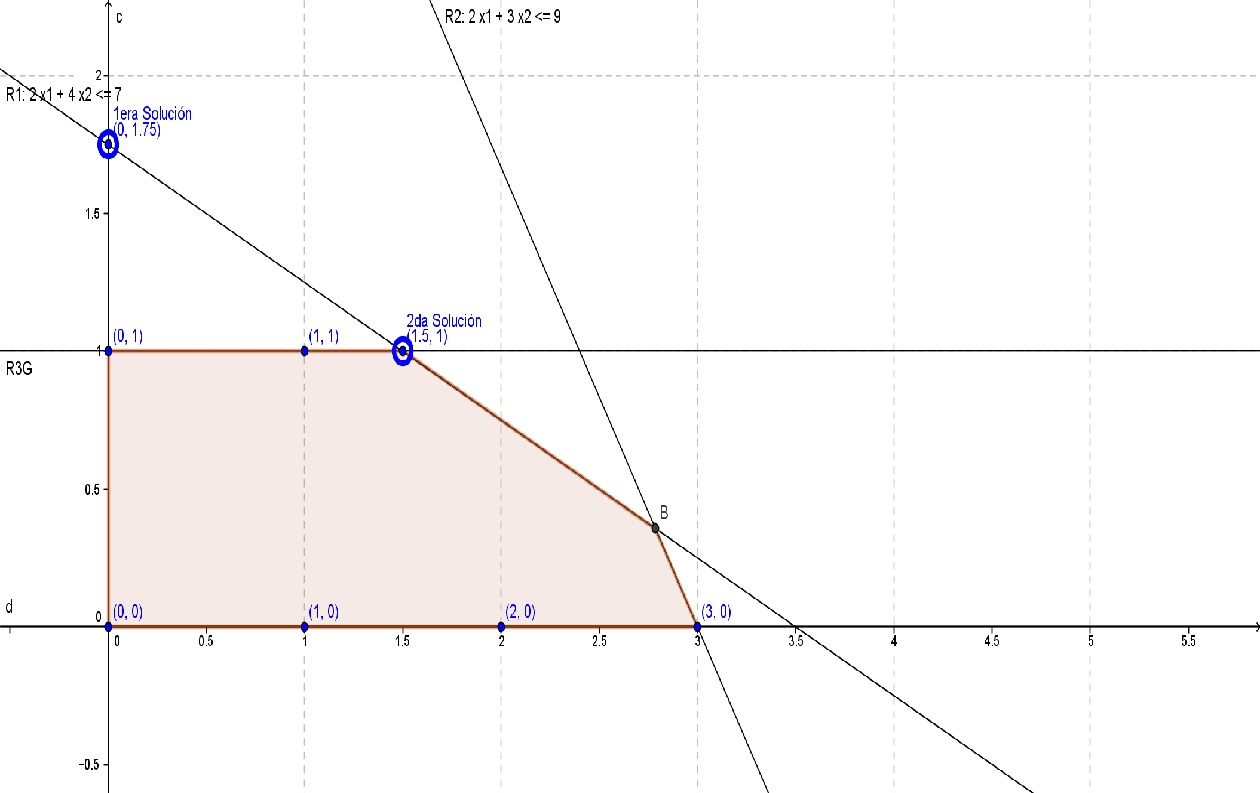
**### Gráfico.**

**### 1era Solución del Problema.**

****

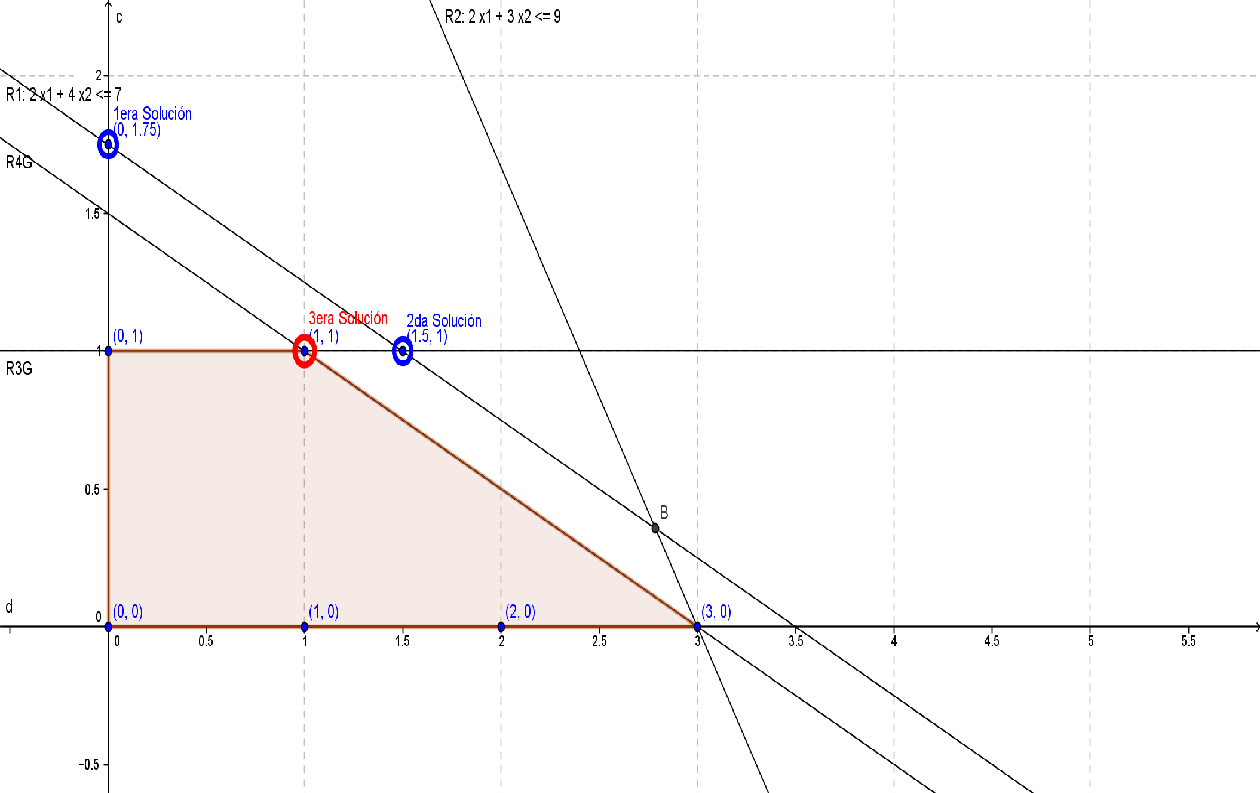
**### Gráfico.**

**### Corte R3G y 2da Solución.**

****

**### Gráfico.**

**### Corte R4G y 3era Solución.**

****

**9. Tercer Ejemplo con Representación Real.**

**==========================================**

**Se tiene el siguiente problema:**

**(0) max z = 5 x1 + 9 x2**

**(1) -1 x1 + 5 x2 <= 3**

**(2) 5 x1 + 3 x2 <= 27**

**(3) x1,x2 >= 0 y enteros**

**Se agregan las variables de holgura:**

**(0) max z = 5 x1 + 9 x2**

**(1) -1 x1 + 5 x2 + 1 s3 = 3**

**(2) 5 x1 + 3 x2 + 1 s4 = 27**

**(3) x1,x2 >= 0 y enteros**

**Al resolver el problema con simplex se obtiene la tabla:**

**>>> Iteración 1:**

**-------------------------------------------------------**

**i BVS x1 x2 s3 s4 RHS**

**-------------------------------------------------------**

**0 z 0.000 0.000 1.071 1.214 36.000**

**1 x2 0.000 1.000 0.179 0.036 1.500**

**2 x1 1.000 0.000 -0.107 0.179 4.500**

**-------------------------------------------------------**

**Al tomar la fila 1 se genera el primer corte de Gomory**

**(1 ) 0.000 x1 + 1.000 x2 + 0.179 s3 + 0.036 s4 = 1.500**

**(3G) 0.179 s3 + 0.036 s4 >= 0.500**

**El problema original se convierte en:**

**(0 ) max z = 5 x1 + 9 x2**

**(1 ) -1 x1 + 5 x2 + 1.000 s3 = 3.000**

**(2 ) 5 x1 + 3 x2 + 1.000 s4 = 27.000**

**(3G) 0.179 s3 + 0.036 s4 - 1.000 s5 = 0.500**

**Al resolver este problema se obtiene la tabla**

**>>> Iteración 2:**

**---------------------------------------------------------------**

**i BVS x1 x2 s3 s4 s5 RHS**

**---------------------------------------------------------------**

**0 z 0.000 0.000 0.000 0.999 5.9999 33.007**

**1 x2 0.000 1.000 0.000 0.000 0.9998 1.001**

**2 x1 1.000 0.000 0.000 0.200 -0.5999 4.799**

**3G s3 0.000 0.000 1.000 0.201 -5.5991 2.793**

**---------------------------------------------------------------**

**De esta tabla se toma de manera arbitraria la restricción 2 y se genera un nuevo corte de Gomory.**

**(2 ) 1.000 x1 + 0.000 x2 + 0.000 x3 + 0.200 s4 - 0.599 s5 = 4.799**

**(4G) 0.200 s4 - 0.401 s5 >= 0.799**

**El problema original se convierte en:**

**(0 ) max z = 5 x1 + 9 x2**

**(1 ) -1 x1 + 5 x2 + 1.000 s3 = 3.000**

**(2 ) 5 x1 + 3 x2 + 1.000 s4 = 27.000**

**(3G) 0.179 s3 + 0.036 s4 - 1.000 s5 = 0.500**

**(4G) 0.200 s4 - 0.401 s5 - 1.000s6 = 0.799**

**Al resolver este problema se obtiene la siguiente tabla:**

**>>> Iteración 3:**

**------------------------------------------------------------------------**

**i BVS x1 x2 s3 s4 s5 s6 RHS**

**------------------------------------------------------------------------**

**0 z 0.000 0.000 0.000 0.000 3.988 4.994 29.017**

**1 x2 0.000 1.000 0.000 0.000 0.997 -0.001 1.002**

**2 x1 1.000 0.000 0.000 0.000 -0.197 1.001 4.000**

**3G s3 0.000 0.000 1.000 0.000 -5.183 1.006 1.990**

**4G s4 0.000 0.000 0.000 1.000 2.005 -5.000 3.995**

**------------------------------------------------------------------------**

**Podemos observar que todavía no se tiene una solución entera. Sin embargo esto es debido a los errores de redondeo. Si se realiza el mismo ejercicio con fracciones se obtendrá el resultado:**

**x1 = 4**

**x2 = 1**

**z = 29**

**10. Algunos Cuidados al Utilizar Cortes.**

**========================================**

**Cuando se utiliza el algoritmo de cortes, es bueno tomar en cuenta las siguientes recomendaciones.**

**Si el nuevo corte que se agrega utiliza nuevas variables de holgura o de excedente, estas deberán incluirse en la tabla, tal como se hizo en el último ejemplo. De no utilizarlas, no es necesario incluirlas. Las variables artificiales no deben incluirse nunca, pues la primera fase del simplex las elimina.**

**Si se utiliza el método de la gran M, en el momento de construir el corte, primero se desechan las variables artificiales y luego se construye el corte con las variables de holgura restantes.**

**El algoritmo de cortes de Gomory suele converger rápidamente en muchas ocasiones, sin embargo pueden ocurrir convergencias lentas, donde las soluciones se aproximan a los valores enteros poco a poco.**

**Al utilizar notación de punto flotante, puede ocurrir que el algoritmo no converga debido a los errores de redondeo. En este ejemplo se debe mencionar que si se utilizan pocos dígitos decimales el algoritmo tarda mucho más en converger o no convergerá del todo.**

**Finalmente es importante mencionar que existen varios métodos para poner la nueva restricción de Gomory y continuar a partir de la última tabla del simplex de manera que no hay necesidad de volver a plantear el problema desde el inicio. Los más conocidos de estos métodos son "el simplex-dual" y "la perturbación T de Gillet".**

**11. Resumen.**

**============**

**Se ha investigado un modelo en el cual el supuesto de divisibilidad no se mantiene. Se han modelado problemas enteros y se han presentado aplicaciones de dicho modelo.**

**Se analizó el algoritmo de cortes de planos de Gomory, en el cual se construyen nuevas restricciones las cuales "cortan" el espacio de soluciones factibles, de manera que el algoritmo del simplex converge hacia una solución entera.**

**Este algoritmo fue hecho para trabajar con representación númerica de fracciones, al utilizar punto flotante puede ocurrir que el algoritmo no converga.**

**+++**

**12. Ejercicios.**

**===============**

**() Ejercicio 1.**

**Indique las diferencias fundamentales entre un problema de programación lineal entera y uno continuo.**

**() Ejercicio 2.**

**Indique los pasos que lleva el algoritmo de Gomory.**

**Investigue si existe algún otro algoritmo de corte.**

**() Ejercicio 3.**

**Resuelva el siguiente problema de programación entera.**

**max z = 7 x1 + 9 x2**

**-1 x1 + 3 x2 <= 6**

**7 x1 + 1 x2 <= 35**

**x1,x2 >= 0 y enteras.**

**() Ejercicio 4.**

**Resuelva el siguiente problema de programación entera.**

**max z = 1 x1 + 2 x2**

**2 x2 <= 7**

**1 x1 + 1 x2 <= 7**

**2 x1 <= 11**

**x1,x2 >= 0 y enteras.**

**() Ejercicio 5.**

**Resuelva el siguiente problema de programación entera.**

**max z = 2 x1 + 6 x2**

**3 x1 + 1 x2 <= 5**

**4 x1 + 4 x2 <= 9**

**x1,x2 >= 0 y enteros**

**() Ejercicio 6.**

**Resuelva el siguiente problema de programación entera.**

**max z = 5 x1 + 9 x2**

**-1 x1 + 5 x2 <= 3**

**5 x1 + 3 x2 <= 27**

**x1,x2 >= 0 y enteras**

**() Ejercicio 7.\***

**Resuelva el siguiente problema de programación entera.**

**max z = 2 x1 + 1 x2**

**2 x1 + 5 x2 <= 17**

**3 x1 + 2 x2 <= 10**

**x1,x2 >= 0 y enteras.**

**() Ejercicio 8.**

**Resuelva el siguiente problema de programación entera.**

**Es algo largo pues toma como 6 iteraciones resolverlo.**

**min z = 20 x1 + 22 x2 + 18 x3**

**4 x1 + 6 x2 + 1 x3 >= 54**

**4 x1 + 4 x2 + 6 x3 >= 65**

**1 x1 <= 7**

**1 x2 <= 7**

**1 x3 <= 7**

**x1,x2,x3 >= 0 y enteras**

**() Ejercicio 9.**

**Resuelva el siguiente problema de programación entera.**

**max z = 3 x1 + 4 x2**

**2 x1 + 1 x2 <= 6**

**2 x1 + 3 x2 <= 9**

**x1,x2 >= 0 y enteras.**

**() Ejercicio 10.**

**Resuelva el siguiente problema de programación entera.**

**max z = 1 x1 + 9 x2 + 1 x3**

**1 x1 + 2 x2 + 3 x3 <= 9**

**3 x1 + 2 x2 + 2 x3 <= 15**

**x1,x2,x3 >= 0 y enteras.**

**() Ejercicio 11: Cortes de Dantzig \*\*\***

**George Dantzig, propuso inicialmente una forma de cortes para solucionar los problemas de programación entera. La idea de este algoritmo se basa en el hecho que al menos una de las variables no básicas debe ser mayor que cero para obtener una solución entera.**

**De esta forma en cada problema se agrega la suma de las variables no básicas de forma que sean mayores o iguales a 1, de la siguiente manera:**

**vk[1] + vk[2] + ... + vk[n] >= 1**

**Donde cada variable vk[i] es no básica.**

**Los cortes cambiarán en cada iteración porque el conjunto de variables de no básicas cambiarán también.**

**Resuelva el siguiente problema utilizando los cortes de Dantzig:**

**(0) max z = 3 x1 + 4 x2**

**(1) 2 x1 + 1 x2 <= 6**

**(2) 2 x1 + 3 x2 <= 9**

**(3) x1,x2 >= 0 y enteros**

**Para resolver este problema necesitará agregar 3 cortes de Dantzig. A continuación se esbozan los resultados que se esperan obtener:**

**Iteración 0:**

**Del problema original obtendrá:**

**x1 = 2.25**

**x2 = 1.50**

**Iteración 1:**

**Al agregar la 1ra restricción de Dantzig:**

**x1 = 1.50**

**x2 = 2.00**

**Iteración 2:**

**Al agregar la 2da restricción de Dantzig:**

**x1 = 0.75**

**x2 = 2.50**

**Iteración 3:**

**al agregar la 3ra restricción de Dantzig:**

**x1 = 0.00**

**x2 = 3.00**

**z = 12.00**