

Solucion de problemas por metodo Dual Simplex.

Alma Hernandez-Flores
Instituto Tecnológico Superior Zacatecas Occidente

INTRODUCCIÓN.

El uso de las herramientas tecnológicas facilita las técnicas de solución para problemas de programación lineal, en el siguiente trabajo describiré los pasos para la solución problemas de programación lineal empleando la herramienta “AMPL”.

METODOLOGÍA.

Se resolverán problemas de programación lineal mediante el uso del programa “AMPL”

PROBLEMA DISTRIBUCIÓN DE HORAS DE TRABAJO Y PING PONG.

Asume que quieres decidir entre formas alternas de pasar 1 día de 8 horas, esto es que se quiere distribuir el tiempo. Asume que se te hace 5 veces mas divertido jugar ping pong que trabajar, pero también sientes que debes trabajar por lo menos 3 veces tantas horas como las que jugaste ping pong. El problema es saber cuantas horas debes jugar y cuantas horas trabajar y maximizar tu objetivo (diversión)

Variables de decisión.

X = numero de horas trabajando

Y= numero de horas jugando

```
<Maximizar: f= X + 5Y
Sujeto a:
c1: x+y<=8 horas
c2: 3y<=x
c3: x>=0
c4: y>=0>
```

Solución del problema por medio de AMPL

Primero se deben identificar las variables de decisión.:

Una de nuestras variables es el numero de horas trabajando y la otra el numero de horas jugando, y asi es como se escriben en el programa de AMPL.

```
<ampl: var trabajando;
ampl: var jugando;>

:
```

Después se define nuestra función en base a la diversión, esta se conforma a partir de que se disfruta 5 veces mas jugar ping pong que trabajar.:

```
<ampl: maximize diversion: 1*trabajando+5*jugando>
```

Posteriormente se identifican las restricciones las cuales están sujetas a la función objetivo.:

```
<ampl: subject to
ampl? c1: trabajando+jugando<=8;
ampl: c2: 3*jugando<=trabajando;
ampl: c3: trabajando>=0;
ampl: c4: jugando>=0;>
```

:

Una vez que están definidas las constricciones, se determina la solución optima. : Esto se hace tecleando la palabra “solve” y automáticamente el Programa de AMPL encontrara la solución optima

```
<ampl: solve;
MINOS 5.51: optimal solution found.
1 iterations, objective 16>
```

Para que el programa de a conocer las cantidades exactas de las variables del problema solo basta con teclear la palabra “display” junto con todas las variables, y por si solo el programa desplegara las cantidades de cada una de las variables.

```
<ampl: display diversion, trabajando, jugando;
diversion = 16
trabajando = 6
jugando = 2>
```

En base a los resultados se deberán jugar 6 horas y trabajar dos horas, esto con el fin de maximizar el tiempo.

Solución general del problema en AMPL. :

```
<ampl: var trabajando;
ampl: var jugando;
ampl: maximize diversion: 1*trabajando+5*jugando;
ampl: subject to
ampl? c1: trabajando+jugando<=8;
ampl: c2: 3*jugando<=trabajando;
ampl: c3: trabajando>=0;
ampl: c4: jugando>=0;>
```

```

ampl: solve;
MINOS 5.51: optimal solution found.
1 iterations, objective 16
ampl: display diversion, trabajando, jugando;
diversion = 16
trabajando = 6
jugando = 2

```

PROBLEMA DE MAXIMIZACION DE LA GANANCIA "PUESTO DE BEBIDAS"

Un muchacho quiere abrir un puesto de bebidas, su mama le dice que no puede vender mas de 4 galones de bebida. El muchacho vende limonadas y jugo de frutas.

Vende la limonada a \$2 dolares el galón y el jugo de fruta a \$1.50 dolares el galón, la limonada requiere 30 rebanadas de limón por galón y 1 libra de azúcar por galón. El jugo de fruta usa 10 rebanadas y 2 libras de azúcar por galón. la mama del muchacho tiene solamente 90 rebanadas de limón y 6 libras de azúcar.

Encontrar cuantos galones de cada bebida se pueden hacer para ser la mayor cantidad de dinero o la mejor ganancia.

<Funcion: $2x + 1.5y$ >

Variables de decisión

X = numero de galones de limonada

Y=numero de galones de jugo de fruta

```

<ampl: var limonada;
ampl: var jugo;>

```

:

Se define la función objetivo, es decir la función que se va a maximizar, sujeta a todas las restricciones :

```

<Maximizar: f= 2x+1.5y
sujeto a:
c1: x+y<=4
c2: 30x+10y<=90
c3: x>=0
c4: y>=0
c5: x+2y<=6>

```

posteriormente se encuentra la solución optima.:

```

<ampl: solve;
MINOS 5.51: optimal solution found.
2 iterations, objective 7.25>

```

:

Se identifican las cantidades exactas de cada una de las variables.:

```

<ampl: display ganancia, limonada, jugo;
ganancia = 7.25
limonada = 2.5
jugo = 1.5>

```

Basándose en los resultados se deben hacer 2.5 galones de limonada Y 1.5 galones de jugo de fruta, para obtener una ganancia de 7.25.

Solución general en el programa AMPL:

```

<ampl: var limonada;
ampl: var jugo;
ampl: maximize ganancia: 2*limonada+1.5*jugo;
ampl: subject to
ampl? c1: 1*limonada+1*jugo<=4;
ampl: c2: 30*limonada+10*jugo<=90;
ampl: c3: 1*limonada+2*jugo<=6;
ampl: c4: limonada>=0;
ampl: c5: jugo>=0;
ampl: solve;
MINOS 5.51: optimal solution found.
2 iterations, objective 7.25
ampl: display ganancia, limonada, jugo;
ganancia = 7.25
limonada = 2.5
jugo = 1.5>

```

CONCLUSIÓN.

Queda claro que con la utilización de programas como lo es AMPL , es mas fácil realizar o encontrar la solución de problemas matemáticos, con los metodos mas complejos.