Practica 2

Valoración a través de estimar el precio sombra

a) Toma en cuenta el costo de oportunidad del uso de un recurso (agua) que se puede usar como insumo en muchas actividades económicas.
b) Un recurso tiene un precio sombra si es escaso.

Estimar el precio sombra no resuelve el problema de valoración del recurso agua y sus servicios ambientales, ya que, para poder tener un costo de oportunidad, tenemos que contar *a priori* con las alternativas valoradas.

Programación lineal: ejemplo 1

Ejemplos de problemas de programación lineal[1]

1. Roberto es agricultor, tiene 625 acres de tierra y 1000 acre-píes de agua disponibles. También solo puede esperar unas 300 horas a la semana de trabajo (jornaleros). En la siguiente tabla se resumen los componentes de costos e ingreso para cada cultivo que puede plantar:

	Maíz = x	**Trigo = y**	**Avena = z**
Riego (píes)	3	1.0	1.5
Trabajo (horas/semana)	0.8	0.2	0.3
Ingreso (dólares/acre)	400	200	250

La función objetivo de Roberto es **maximizar sus ingresos** algebraicamente podemos definir la función objetivo como:

p(x,y,z) = 400 x + 200 y + 250 z

(es una función lineal ya que ningún termino de la ecuación esta elevado a alguna potencia diferente de 1, los cambios en el ingreso son constantes según el cambio de las variables x, y, z con sus respectivos coeficientes)

x, y z son las variables que representan el número de acres de tierra dedicados a cada una de las tres posibles actividades que se incluyen en el modelo, la suma de estas tres variables tiene que se menor o igual a la superficie disponible 625 acres, que es la restricción de tierra. Algebraicamente puede quedar:

$$x + y + z \leq 625$$

La parte amarilla de la inecuación se le llama el lado izquierdo y a la parte verde el lado derecho, en este último es donde están las restricciones.

[1] Ejemplo tomado de: http://www.uccs.edu/~rcarlson/math448/lin1.pdf (visitado 17 de Agosto del 2015), adaptado y traducido por el Alan Navarro.

La restricción para agua es de 1000 acre-píes queda al lado derecho de la inecuación, en este caso 3x significa que por cada acre de maíz (x) se utilizaran 3 acre-píes de agua, por cada acre dedicado a “y” esto es, trigo, se utilizaría 1 acre-píe de agua, por último por cada acre sembrado de avena, se estaría utilizando 1.5 acre-píes de agua, la sumatoria de estos tres componentes del lado izquierdo de la inecuación nos darían el uso total de agua.

$$3x + y + 1.5\,z \leq 1000$$

Por último, la restricción para el uso del trabajo quedaría:

$$0.8x + 0.2y + 0.3\,z \leq 300$$

Un modelo de programación lineal (PL) bien planteado permite encontrar la solución que: maximiza el ingreso y no viola las restricciones del modelo.
Modelos con pocas variables podrían resolverse manualmente (con lápiz y papel), pero la gran mayoría de los modelos en la vida real involucran muchas variables y muchas restricciones. No es objeto de este curso el resolver gráfica y algebraicamente modelos de PL; lo importante es 1) saber plantearlos y 2) ingresarlos y resolverlos con la ayuda de un paquete computacional.

Resolver en R Studio

Una opción para resolver el problema de PL es usar un software de código abierto (gratuito) conocido como R, en su interface conocida como R Studio. El reto es:

1) Pasar el modelo de su forma algebraica al lenguaje de R.
2) Correr el modelo.
3) Interpretar los resultados.

```r
# Ejemplo 1 Roberto el agricultor
library(linprog) # cargo el programa
# 1. coeficientes de la funcion objetivo c
cvec <- c(400, 200, 250)
names(cvec) <- c("Maiz","Trigo","Avena") #asigno nombres al vector
cvec # para ver el vector creado
#################################################################
## 2. Elaboro un vector con el lado derecho de las
## restricciones
bvec <- c(625, 1000, 300)  # restricción
names(bvec) <- c("Tierra","Agua","Trabajo")
#################################################################
## 3. creo una matriz con los coeficientes técnicos del lado
## izquierdo de las restricciones
Amat <- rbind( c( 1, 1, 1 ), # tierra
               c(3,1, 1.5),  # agua
               c(0.8, 0.2, 0.3) # trabajo
               )
#################################################################
# 4. Vector que determina la dirección de las restricciones <=, >= o =
dir <- rep('<=', 3)
#################################################################
# Determinar si queremos maximizar o minimizar (maximum=TRUE o maximum=FALSE).
## Resuelvo para maximizar el ingreso
res <- solveLP( cvec, bvec, Amat, TRUE, dir )
res
summary(res)
```

```
Results of Linear Programming / Linear Optimization

Objective function (Maximum): 162500

Iterations in phase 1: 0
Iterations in phase 2: 2
Solution
           opt
Maiz   41.6667
Trigo   0.0000
Avena 583.3333

Basic Variables
               opt
Maiz       41.6667
Avena     583.3333
S Trabajo  91.6667

Constraints
          actual dir bvec    free dual dual.reg
Tierra   625.000  <=  625  0.0000  100 291.6667
Agua    1000.000  <= 1000  0.0000  100  62.5000
Trabajo  208.333  <=  300 91.6667    0  91.6667

All Variables (including slack variables)
               opt cvec min.c max.c marg marg.reg
Maiz       41.6667  400   250   500   NA       NA
Trigo       0.0000  200  -Inf   200    0  437.500
Avena     583.3333  250   200   400   NA       NA
S Tierra    0.0000    0  -Inf   100 -100  291.667
S Agua      0.0000    0  -Inf   100 -100   62.500
S Trabajo  91.6667    0  -500   300    0       NA
```

En la sección de restricciones (Constraints) el valor “dual” (precio sombra) nos dice cuánto sería el cambio en el valor de la función objetivo sí cambia en una unidad el lado derecho de la restricción. La columna que dice “dual.reg” nos dice el rango valido para el precio sombra. Por ejemplo, obtener una unidad más de agua aumenta ingreso en 100, pero éste precio sombra sería valido, ceteris paribus, hasta 62.5 unidades, después las condiciones del modelo cambian.

En la tabla **“All Variables”** tenemos varias columnas, la primera nos da, para el caso de las actividades el “nivel de la actividad” óptimo, para el caso de los recursos, los da la “variable de holgura” o “slack variable” es decir ¿sobra recurso? en el caso de la tierra y el agua no sobra recurso, esto es no hay holgura. La segunda columna nos ofrece el coeficiente de cada variable en la función objetivo, el modelo crea un rango de precios en los cuales la solución óptima se mantiene, para el caso del maíz sería 250 <—> 400 <—>500

Los modelos de programación lineal son útiles para hacer lo que se conoce como estática comparada, ésta consiste en correr el modelo sucesivamente haciendo cambios y comparado los valores óptimos; por ejemplo ¿qué pasaría si quisiéramos obligar al modelo a sembrar 20 acres de trigo?

<u>Los resultados del modelo de arriba nos enseñan:</u>

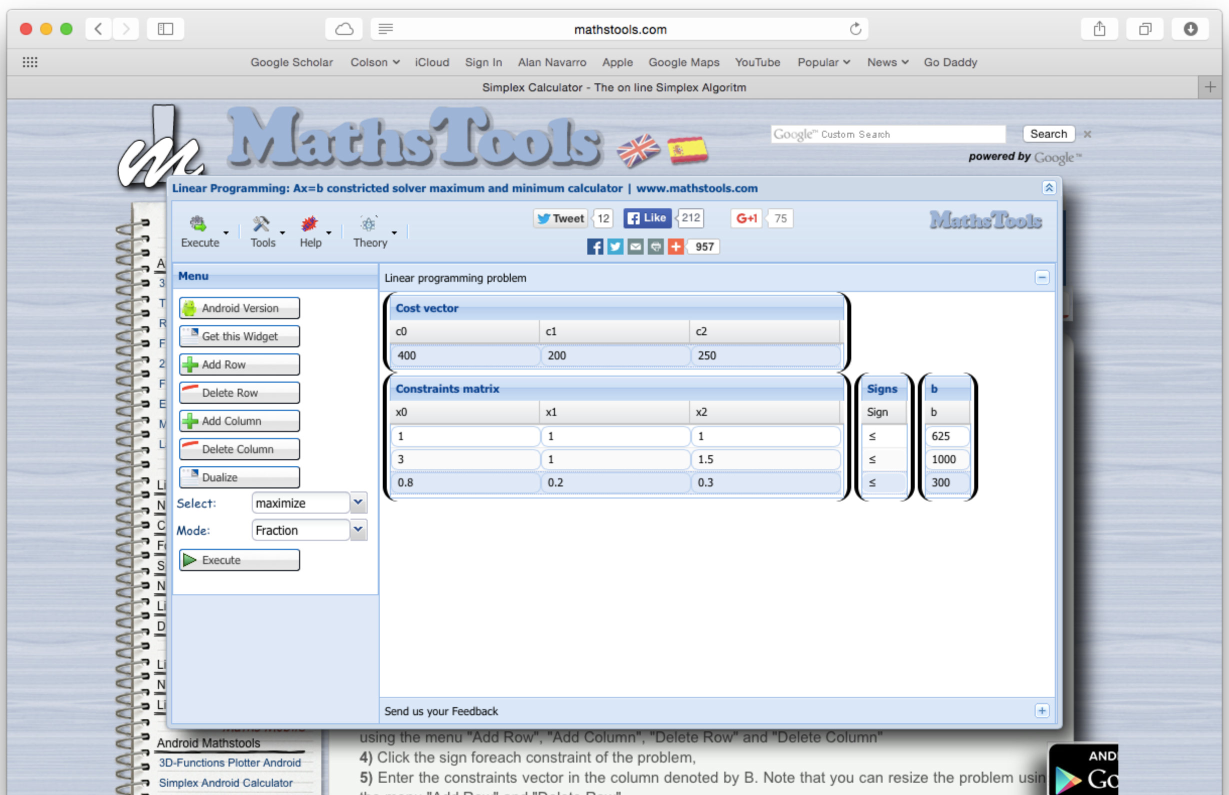
1. El Ingreso se maximiza en $162,500.00
2. Se recomienda sembrar 41.6 acres de maíz.
3. Se recomienda sembrar 583.3 acres de avena.
4. Se utiliza toda la tierra y toda el agua.
5. Aumentar un acre más de tierra, aumenta el valor en $100.
6. **Igualmente, el valor del agua, de un acre-píe (1233.48 metros cúbicos) es de $100.**

R no es el único “solver” para programación lineal, hay muchos otros como por ejemplo Excel, e incluso algunos disponibles en línea en Internet:

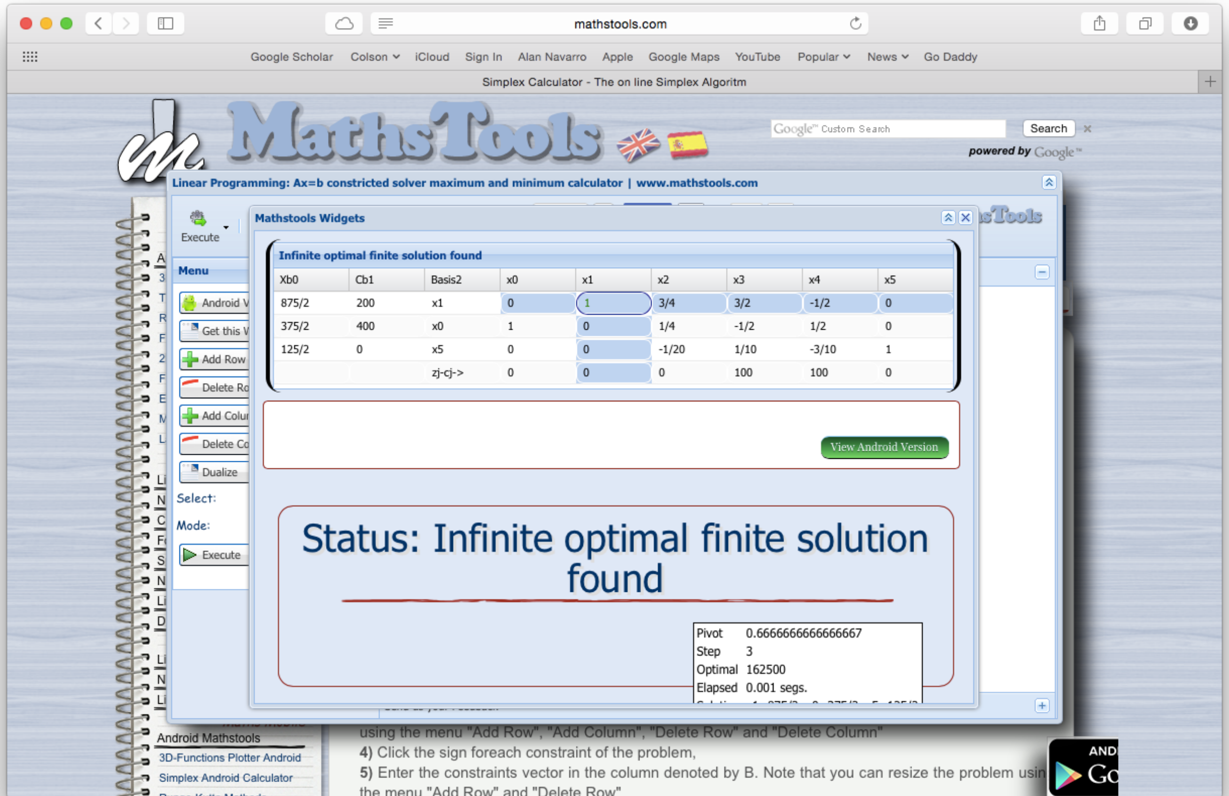
Nuestro modelo es el siguiente:

$$\text{Maximize } p = 400x + 200y + 250z \text{ subject to } x + y + z \le 625,\ 3x + y + 1.5z \le 1000,\ 0.8x + 0.2y + 0.3z \le 300$$

<u>http://www.mathstools.com/section/main/simplex_online_calculator#</u>



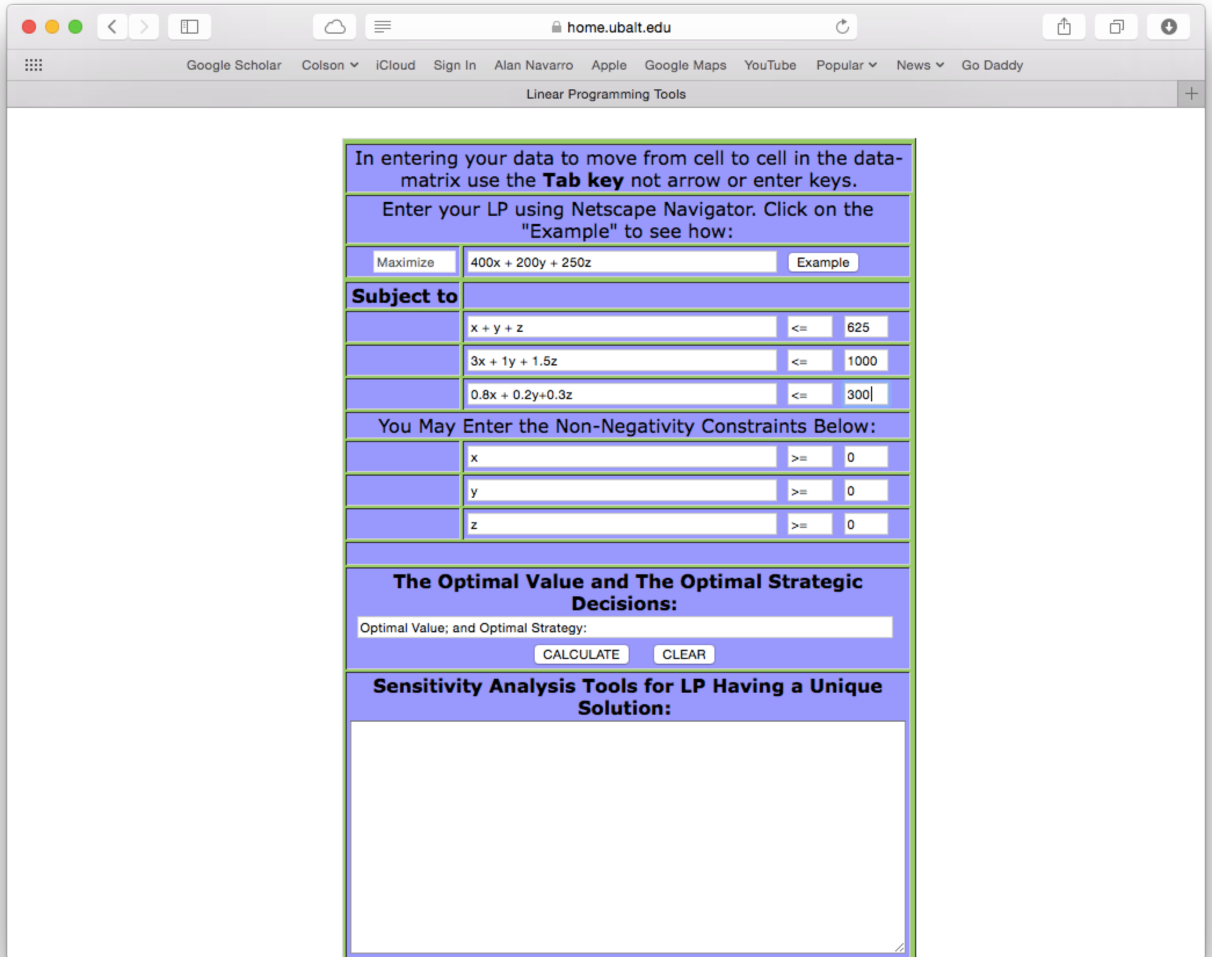
Click en “Execute” y luego “Go to End” (botones verdes), y deberá de aparecer el resultado:

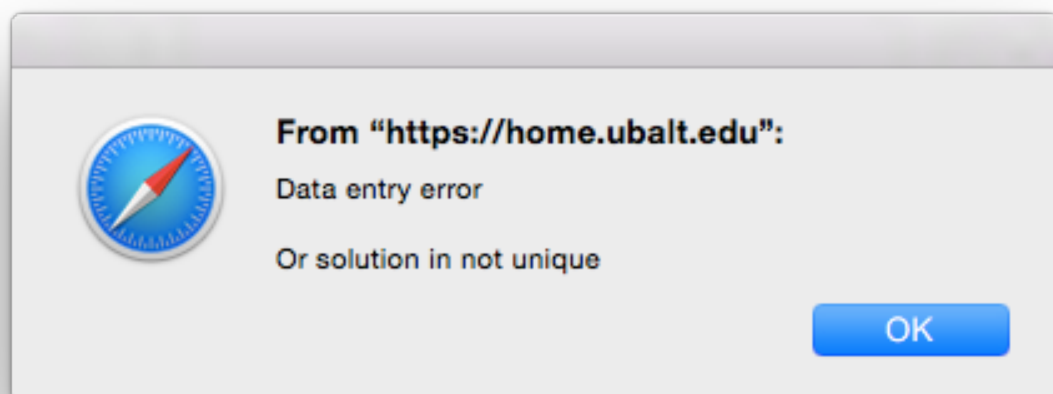


¿Qué pasó? Aparentemente el modelo no tiene una única solución, no es qué la técnica cómo tal no funcione, sino que el problema está mal diseñado.

Usamos otro "Solver" online para ver que pasa.

https://home.ubalt.edu/ntsbarsh/business-stat/otherapplets/LPTools.htm


home.ubalt.edu
Google Scholar
Colson
iCloud
Sign In
Alan Navarro
Apple
Google Maps
YouTube
Popular
News
Go Daddy
Linear Programming Tools
In entering your data to move from cell to cell in the data-matrix use the Tab key not arrow or enter keys.
Enter your LP using Netscape Navigator. Click on the "Example" to see how:
Maximize
400x + 200y + 250z
Example
Subject to
x + y + z
<=
625
3x + 1y + 1.5z
<=
1000
0.8x + 0.2y+0.3z
<=
300
You May Enter the Non-Negativity Constraints Below:
x
>=
0
y
>=
0
z
>=
0
The Optimal Value and The Optimal Strategic Decisions:
Optimal Value; and Optimal Strategy:
CALCULATE
CLEAR
Sensitivity Analysis Tools for LP Having a Unique Solution:


From "https://home.ubalt.edu":
Data entry error
Or solution in not unique
OK

Obviamente este ejemplo no sería el mejor para usar en un curso de introducción a la técnica de programación lineal, mucho menos en un curso que ni siquiera es de “investigación de operaciones” o de “programación matemática”, no obstante, decidí dejarlo ya que enseña una situación común, modelos en cuyo diseño no se puede encontrar una solución única óptima.

Como se puede ver en este tutorial. La solución es simple: no te quedes con el resultado que te da R y haz un “double-check” con otros software.

Buscar un máximo o un mínimo es como ir por una montaña tratando de encontrar el punto más alto o más bajo respectivamente, cuando sabemos que sólo hay un punto que es el más alto de todos, decimos que tenemos un máximo global (o mínimo global), eso es lo que quisiéramos encontrar en un modelo de programación lineal.

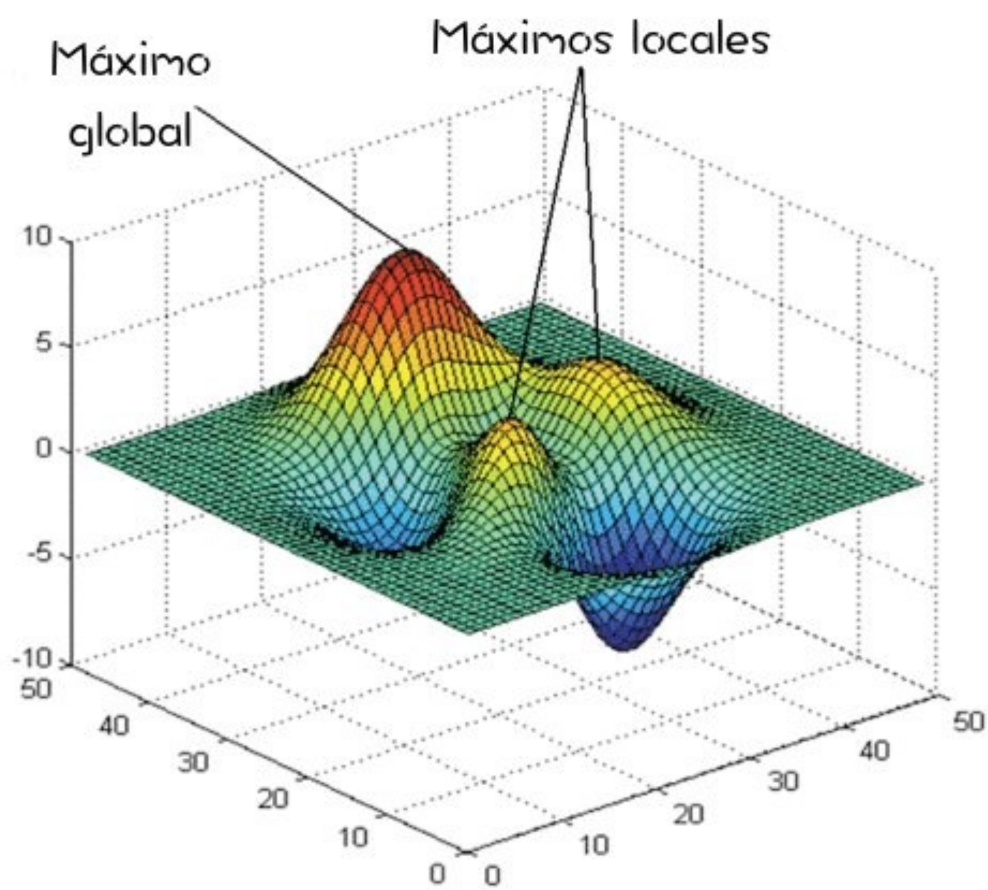


Imagen tomada de: http://www.datingforeconomists.com/blog/2010/03/local-vs-global-maximum.html