

# Programación lineal

Karen Guadalupe Lechuga Trejo

April 12, 2019

## Contents

<b>1</b>	<b>Teoría</b>	<b>1</b>
1.1	Motivación . . . . .	1
1.2	Ejemplos . . . . .	2
1.3	Convexidad . . . . .	3
1.4	Método SIMPLEX . . . . .	3
<b>2</b>	<b>Herramientas computacionales</b>	<b>3</b>
2.1	Emacs . . . . .	3
2.2	Git . . . . .	3
2.3	Python . . . . .	3
2.4	L <sup>A</sup> T <sub>E</sub> X . . . . .	4

## 1 Teoría

### 1.1 Motivación

El objetivo de la programación lineal es maximizar funciones lineales sobre dominios convexos, es decir, definidos sobre regiones dadas por desigualdades.



## 1.2 Ejemplos

- El problema de la dieta.
- Optimización de lugares en una excursión.
- Escoger objetos óptimos para un campamento.

- El problema del flujo máximo.

### 1.3 Convexidad

Un conjunto  $X$  es **convexo** si para todos  $x, y \in X$  y  $t \in [0, 1]$  se tiene que  $tx + (1 - t)y \in X$ .

### 1.4 Método SIMPLEX

## 2 Herramientas computacionales

### 2.1 Emacs

C-x C-s	salvar archivo
C-x C-f	abrir archivo
M-q	formatear párrafo
C-x d	editar directorios
C-g	interrumpe procesos
C-x 1	regresa a una sola pantalla
C-x 2	divide horizontalmente
C-x 3	divide verticalmente
M-w	copiar la región
C-w	borrar la región
shift-flechas	seleccionar la región
C-y	pegar la región
C-c C-e	menú exportar en otros formatos
M-flechas	mover renglones/columnas de tabla

#### 1. Org mode

C-c C-c	corre un bloque de código
C-x b	cambiar buffer

### 2.2 Git

#### 1. Github

### 2.3 Python

#### 1. Lenguaje Python

En el lenguaje Python podemos hacer operaciones:

2+2

También podemos usar la biblioteca pulp.

```
from pulp import *
x = LpVariable("x", 0, 3)
y = LpVariable("y", 0, 1)
prob = LpProblem("myProblem", LpMinimize)
prob += x + y <= 2
prob += -4*x + y
status = prob.solve()
value(x), value(y), value(prob.objective)
```

2. Jupyter

## 2.4 $\LaTeX$