Rebeca Jaramillo Camarillo

Matricula: 2132988

Materia: Investigación de operaciones **Grupo: 032**

FCFM
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

A3 - Modelado matemático

Problema 4. Cuatro personas (Amy, Jim, John y Kelly) desean cruzar un río utilizando una canoa que sólo puede llevar a dos personas a la vez. Amy puede remar en 1 minuto, Jim en 2 minutos, John en 5 minutos y Kelly en 10 minutos. Si dos personas cruzan juntas, el tiempo está determinado por la persona más lenta.

a) Identifique al menos dos planes factibles para cruzar el río.
b) Defina el criterio para evaluar las alternativas.
c) Determine el menor tiempo necesario para que las cuatro personas crucen el río.

Parámetros

- Tiempo de Amy: 1 minuto
- Tiempo de Jim: 2 minuto
- Tiempo de John: 5 minuto
- Tiempo de Kelly: 10 minuto
- Si dos personas cruzan juntas, el tiempo está determinado por la persona más lenta.

Variables de decisión

$x_{ij} = indica\ si\ una\ persona\ i\ cruza\ el\ rio\ con\ una\ persona\ j$

- $x_{A,Ji} = \{1\ si\ Amy\ cruza\ con\ Jim; 0\ si\ no\}$
- $x_{A,Jo} = \{1\ si\ Amy\ cruza\ con\ John; 0\ si\ no\}$
- $x_{A,K} = \{1\ si\ Amy\ cruza\ con\ Kelly; 0\ si\ no\}$
- $x_{Ji,Jo} = \{1\ si\ Jim\ cruza\ con\ John; 0\ si\ no\}$
- $x_{Ji,K} = \{1\ si\ Jim\ cruza\ con\ Kelly; 0\ si\ no\}$
- $x_{Jo,K} = \{1\ si\ John\ cruza\ con\ Kelly; 0\ si\ no\}$
- $x_{A,regresa} = \{1\ si\ Amy\ regresa; 0\ si\ no\}$

- $x_{Ji,regresa} = \{1\ si\ Jim\ regresa; 0\ si\ no\}$
- $x_{Jo,regresa} = \{1\ si\ John\ regresa; 0\ si\ no\}$
- $x_{K,regresa} = \{1\ si\ Kelly\ regresa; 0\ si\ no\}$

Función objetivo

- Minimizar el tiempo de viaje
- $z = 2x_{A,Ji} + 5x_{A,Jo} + 10x_{A,K} + 5x_{Ji,Jo} + 10x_{Ji,K} + 10x_{Jo,K} + x_{A,regresa} + 2x_{Ji,regresa} + 5x_{Jo,regresa} + 10x_{K,regresa}$

Restricciones

- Si dos personas cruzan juntas, el tiempo está determinado por la persona más lenta.
- $\sum_{ij} x_{ij} \leq 2$ (Pueden cruzar un máximo de dos personas a la vez)
- $\sum_{ij} x_{i,regreso} = 2$ (Dos personas deben regresar para que todos puedan cruzar)
    - $x_{A,regresa} + x_{Ji,regresa} + x_{Jo,regresa} + x_{K,regresa} = 2$
- Todos deben de cruzar el rio al menos una vez
    - $x_{A,Ji} + x_{A,Jo} + x_{A,K} \geq 1$ (Amy cruza)
    - $x_{Ji,Jo} + x_{Ji,K} + x_{A,Ji} \geq 1$ (Jim cruza)
    - $x_{Jo,K} + x_{A,Jo} + x_{Ji,Jo} \geq 1$ (John cruza)
    - $x_{A,K} + x_{Ji,K} + x_{Jo,K} \geq 1$ (Kelly cruza)

```
-  from pulp import LpMinimize, LpProblem, LpVariable
-
-  # Crear el problema
-  prob = LpProblem("Cruzar_el_rio", LpMinimize)
-
-  # Variables de decisión
-  x_AJi = LpVariable("Amy cruza con Jim", 0, 1, cat="Binary")
-  x_AJo = LpVariable("Amy cruza con John", 0, 1, cat="Binary")
-  x_AK = LpVariable("Amy cruza con Kelly", 0, 1, cat="Binary")
-  x_JiJo = LpVariable("Jim cruza con John", 0, 1, cat="Binary")
-  x_JiK = LpVariable("Jim cruza con Kelly", 0, 1, cat="Binary")
-  x_JoK = LpVariable("John cruza con Kelly", 0, 1, cat="Binary")
-  x_A_regresa = LpVariable("Amy regresa", 0, 1, cat="Binary")
-  x_Ji_regresa = LpVariable("Jim regresa", 0, 1, cat="Binary")
-  x_Jo_regresa = LpVariable("John regresa", 0, 1, cat="Binary")
-  x_K_regresa = LpVariable("Kelly regresa", 0, 1, cat="Binary")
```

```
-
-   # Función objetivo
-   prob += (
-       2 * x_AJi
-       + 5 * x_AJo
-       + 10 * x_AK
-       + 5 * x_JiJo
-       + 10 * x_JiK
-       + 10 * x_JoK
-       + 1 * x_A_regresa
-       + 2 * x_Ji_regresa
-       + 5 * x_Jo_regresa
-       + 10 * x_K_regresa
-   )
-
-   # Restricciones
-   prob += x_AJi + x_AJo + x_AK == 1  # Amy cruza
-   prob += x_AJi + x_JiJo + x_JiK == 1  # Jim cruza
-   prob += x_AJo + x_JiJo + x_JoK == 1  # John cruza
-   prob += x_AK + x_JiK + x_JoK == 1  # Kelly cruza
-   prob += x_A_regresa + x_Ji_regresa + x_Jo_regresa + x_K_regresa == 2  #
    Regreso
-
-   # Resolver el problema
-   prob.solve()
-
-   # Resultados
-   print("Estado:", prob.status)
-   print("Tiempo mínimo para cruzar:", prob.objective.value(), "minutos")
-   for var in prob.variables():
-       print(var.name, "=", var.value())
```

```
Tiempo mínimo para cruzar: 15.0 minutos
Amy_cruza_con_Jim = 1.0
Amy_cruza_con_John = 0.0
Amy_cruza_con_Kelly = 0.0
Amy_regresa = 1.0
Jim_cruza_con_John = 0.0
Jim_cruza_con_Kelly = 0.0
Jim_regresa = 1.0
John_cruza_con_Kelly = 1.0
John_regresa = 0.0
Kelly_regresa = 0.0
```

Amy cruza con Jim – 2 min

Amy regresa – 1 min

Jim cruza con Kelly – 10 min

Jim regresa – 2 min

Amy cruza con Jim – 2 min

Problema 5. En un juego de béisbol, Jim es el lanzador y Joe es el bateador. Jim puede lanzar una bola rápida o una curva al azar. Joe predice el tipo de lanzamiento con diferentes promedios de bateo según la combinación.

Parámetros

- Jim lanza una bola rápida
- Jim lanza una bola curva

Variables de decisión

- $x_1 =$ Jim lanza una bola rápida y Joe predice una bola rápida
- $x_2 =$ Jim lanza una bola rápida y Joe predice una bola curva
- $x_3 =$ Jim lanza una bola curva y Joe predice una bola curva
- $x_4 =$ Jim lanza una bola curva y Joe predice una bola rápida

Función objetivo

- Minimizar las predicciones de bateo (asumir que hay un 50% de probabilidad de cualquier cosa)
- $z = 0.5x_1 + 0.5x_2 + 0.5x_3 + 0.5x_4$

Restricciones

- $x_1 + x_2 = 1$ (probabilidad de lanzar una bola rápida)
- $x_3 + x_4 = 1$ (probabilidad de lanzar una bola curva)
- $x_1 + x_4 = 1$ (probabilidad de predecir una bola rápida)
- $x_2 + x_3 = 1$ (probabilidad de predecir una bola curva)
- $0 \leq x_1, x_2, x_3, x_4 \leq 1$

a) Defina las alternativas para este caso.

Jim lanza una bola rápida y Joe predice una bola rápida

Jim lanza una bola rápida y Joe predice una bola curva

Jim lanza una bola curva y Joe predice una bola curva

Jim lanza una bola curva y Joe predice una bola rápida

b) Determine la función objetivo para el problema y explique cómo difiere de la optimización común (maximización o minimización) de un criterio.

Este problema ser de maximización o de minimización dependiendo de cual perspectiva, es decir, de cual jugador se quiere resolver, ya que Jim quiere minimizar las predicciones de bateo de Joe, y Joe quiere maximizar sus predicciones de bateo.

```python
import numpy as np
from scipy.optimize import linprog

# Coeficientes de la función objetivo
c = [0.5,0.5,0.5,0.5]

# Restricciones de igualdad
A_eq = [
    [1, 1, 0, 0],  # probabilidad de lanzar una bola rápida
    [0, 0, 1, 1],  # probabilidad de lanzar una bola curva
    [1, 0, 1, 0],  # probabilidad de predecir bola rápida
    [0, 1, 0, 1],  # probabilidad de predecir curva
]

b_eq = [1, 1, 1, 1]  # Las probabilidades deben sumar 1

# Límites: las probabilidades de cada estrategia deben estar entre 0 y 1
bounds = [(0, 1), (0, 1), (0, 1), (0, 1)]

# Resolver el problema de programación lineal
result = linprog(c, A_eq=A_eq, b_eq=b_eq, bounds=bounds, method='highs')

# Mostrar los resultados
print("Probabilidad para Jim lanzar bola rápida y Joe predecir bola rápida
(x1):", result.x[0])
print("Probabilidad para Jim lanzar bola rápida y Joe predecir bola curva (x2):",
result.x[1])
print("Probabilidad para Jim lanzar bola curva y Joe predecir bola curva (x3):",
result.x[2])
print("Probabilidad para Jim lanzar bola curva y Joe predecir bola rápida (x4):",
result.x[3])
```

```
Probabilidad para Jim lanzar bola rápida y Joe predecir bola rápida (x1): -0.0
Probabilidad para Jim lanzar bola rápida y Joe predecir bola curva (x2): 1.0
Probabilidad para Jim lanzar bola curva y Joe predecir bola curva (x3): 1.0
Probabilidad para Jim lanzar bola curva y Joe predecir bola rápida (x4): 0.0
```