



## Moderando el conflicto interno de opiniones en una red social

Calderón Prieto Brandon (2125974)      Cely Archila Juleipssy Daianne (2122036)  
Fonseca Idarraga Juan David (2323942)

25 de julio de 2025

# Índice general

0.1.	Modelo genérico . . . . .	1
0.1.1.	Parámetros . . . . .	1
0.1.2.	Variables . . . . .	1
0.1.3.	Restricciones . . . . .	1
0.1.4.	Función objetivo . . . . .	1
0.1.5.	Clasificación . . . . .	1
0.2.	Implementación . . . . .	2
0.3.	Análisis de Branch and Bound . . . . .	2
0.3.1.	Descripción del mecanismo . . . . .	2
0.3.2.	Análisis de árboles generados . . . . .	2
0.3.3.	Ejemplos con visualizador de MiniZinc . . . . .	2
0.4.	Instancias y pruebas . . . . .	2
0.4.1.	Instancias de prueba provistas . . . . .	2
0.4.2.	Instancias adicionales generadas . . . . .	2
0.5.	Análisis de resultados . . . . .	2
0.6.	Conclusiones . . . . .	2

## 0.1. Modelo genérico

### 0.1.1. Parámetros

- $n \in \mathbb{N}$ : número total de personas.
- $m \in \mathbb{N}$ : número total de opiniones.
- $p \in \mathbb{N}^m$ : vector con la distribución de personas por opinión, donde  $p_i$  es el número de personas que inicialmente tienen la opinión  $i \in 1 \dots m$ ,  $\sum_{i=1}^m p_i = n$ .
- $e \in [0, 1]^m$ : vector con los valores de extremismo de las opiniones, donde  $e_i \in [0, 1]$  es el valor de extremismo de la opinión  $i \in 1 \dots m$ .
- $c$ : matriz de costes, donde  $c_{i,j} \in \mathbb{R}^+$  es el coste de mover una persona de la opinión  $i$  a la opinión  $j$ , para  $i, j \in 1 \dots m$  ( $c_{i,i} = 0$ ).
- $ce$ : vector de coste extra, donde  $ce_i \in \mathbb{R}^+$  es el coste adicional de mover una persona a la opinión  $i$  si esa opinión estaba inicialmente vacía, para  $i \in 1 \dots m$ .
- $ct \in \mathbb{R}^+$ : coste total permitido.
- $M \in \mathbb{N}$ : número máximo de movimientos permitidos.

### 0.1.2. Variables

Una matriz  $s$ , donde  $s_{i,j} \in \mathbb{N}$  es el número de personas movidas de la opinión  $i$  a la opinión  $j$ , para  $i, j \in 1 \dots m$ . Esta matriz es de dimensiones  $m \times m$  y debe cumplir las siguientes restricciones:

- $\sum_{j=1}^m s_{i,j} = p_i$ : para cada opinión inicial  $i$ , la suma de personas que se mueven desde esa opinión hacia todas las demás (incluida ella misma) debe ser igual al número de personas que originalmente tenían la opinión  $i$ .  $\sum_{i=1}^m \sum_{j=1}^m s_{i,j} = n$ .
- $s_{i,j} \geq 0$  para todo  $i, j \in 1 \dots m$ .

### 0.1.3. Restricciones

#### Numero de movimientos

$$\sum_{i=1}^m \sum_{j=1}^m s_{i,j} \cdot |j - i| \leq M \quad (1)$$

#### Coste total

$$\sum_{i=1}^m \sum_{j=1}^m c_{i,j} \left(1 + \frac{p_i}{n}\right) * s_{i,j} + \delta_{p_j,0} \cdot ce_j * s_{i,j} \leq ct \quad (2)$$

$\delta_{p_j,0}$  es una función indicadora que vale 1 si  $p_j = 0$  y 0 en caso contrario.

### 0.1.4. Función objetivo

La idea es minimizar el extremismo, que se calcula con la siguiente fórmula:

$$E(p', e) = \sum_{i=1}^m p'_i * e_i \quad (3)$$

- $p'$ : vector con la distribución de personas tras aplicar los movimientos de  $s$ .
- $e$ : vector con los valores de extremismo de las opiniones.

### 0.1.5. Clasificación

Aunque todas las variables de decisión son enteras, para modelar las restricciones es necesario usar variables de tipo `float`. Esto hace que el modelo sea un *Programación Lineal Entera Mixta*.

## 0.2. Implementación

Gracias a las instrucciones de modelado de MiniZinc y la naturaleza del problema, las restricciones y la función objetivo se pueden expresar en pocas líneas de código.

La implementación puede describirse en los siguientes pasos:

- Computación de distancias: para evitar el cálculo repetido de las distancias entre opiniones, se crea una matriz  $d$  donde  $d_{i,j} = |i - j|$ .
- Conservación de flujos ( $\sum_{j=1}^m s_{i,j} = p_i$ ): se hace con el fin de que nadie "desaparezca". También se añade la restricción  $s_{i,j} \leq p_i$  para acotar dominios y acelerar la búsqueda.
- Cálculo de la distribución final: para calcular la distribución final se usó la fórmula  $p'[j] = \sum_{i=0}^m s[i, j] \forall j = 1 \dots m$ , personas que "terminan" en la opinión  $j$ .
- Integración con Python: se utiliza la librería minizinc para ejecutar el modelo de MiniZinc desde Python, permitiendo una mayor flexibilidad en la gestión de instancias y resultados.

Obsérvese que `s`, `total_moves` y `p'` son variables enteras, mientras que `total_cost` y `extremism` son continuas. Por ello, el modelo es de tipo *Programación Lineal Entera Mixta*.

## 0.3. Análisis de Branch and Bound

### 0.3.1. Descripción del mecanismo

### 0.3.2. Análisis de árboles generados

### 0.3.3. Ejemplos con visualizador de MiniZinc

## 0.4. Instancias y pruebas

### 0.4.1. Instancias de prueba provistas

### 0.4.2. Instancias adicionales generadas

## 0.5. Análisis de resultados

## 0.6. Conclusiones