



**FACULTAD
DE INGENIERIA**

Universidad de Buenos Aires

71.14 Modelos y Optimización I
1º Cuatrimestre 2022



**FACULTAD
DE INGENIERIA**

Universidad de Buenos Aires

71.14 Modelos y Optimización I

1º Cuatrimestre 2022

Trabajo Práctico - Segunda entrega

Facultad de Ingeniería

Universidad de Buenos Aires

Ponce Matias 89820



Análisis de la situación

Se trata de un problema combinatorio estilo del viajante ya que se cumplen las condiciones:

- 1.- Se vuelve al lugar de origen
- 2.- Todas las sucursales se visitan una vez
- 3.- Ninguna sucursal se visita más de una vez
- 4.- No se conoce el orden y el valor optimo del funcional depende del orden elegido

Se tiene la problemática que no se pueden armar todas las configuraciones posibles y luego elegir la mejor ya que al ser un numero grande sucursales el número de combinaciones es exponencial.

Ideas de cómo lo van a intentar resolver

Primera entrega

Partiendo desde la central se seleccionan los posibles destinos (según capacidad mínima y máxima del camión).

De los posibles destinos se selecciona el mejor (menor distancia)

Luego en cada iteración (para cada sucursal) se buscan los siguientes destinos posibles (según capacidad mínima y máxima del camión. Además, debe ser una sucursal que no se haya visitado antes).

De los posibles destinos se selecciona el mejor (menor distancia).

Al final se muestra el circuito recorrido.

Segunda entrega

Se mantiene lógica inicial usada en la primera entrega pero se aplica algunas mejoras ya que el programa propuesto presentaba errores de memoria **STD::BAD_ALLOC**



Comentarios sobre los cambios

Primera entrega

En un primer momento intenté armar todas las combinaciones posibles y luego quedarme con la mejor, pero vi que esto es inviable dada la cantidad de sucursales. Como se tenía un incremento exponencial se consumía mucho tiempo y memoria.

Segunda entrega

Se presentó el principal problema de **STD::BAD_ALLOC**, que es una excepción arrojada por el sistema por falta de recursos de memoria.

La falta de recursos de memoria se debía a una ineficiente programación que se había propuesto en la primera entrega.

Entre las mejoras propuestas se implementaron:

- Uso de memoria dinámica y liberación.
- Pasaje de variables por referencia en lugar de copias.
- Evitar recorridos en listas innecesarios.
- Evitar recursividad
- No evaluar sucursales ya visitadas innecesariamente.

Objetivo

Determinar el orden en qué visitar las sucursales para poder visitar todas cumpliendo las restricciones del camión para minimizar distancias en el periodo determinado.

Hipótesis y supuestos:

- 1.- Desde la central se puede visitar cualquier sucursal
- 2.- Desde una sucursal se puede visitar cualquier otra sucursal
- 3.- Todas las sucursales deben ser visitadas una única vez
- 4.- Se dispone del tiempo para cumplir con el recorrido determinado por el modelo
- 5.- Se puede ir a la central desde cualquier sucursal



6.- Se toma como ubicación de la sucursal central el (0,0)

7.- El camión sale desde la central sin dinero

Constantes

CANT_SUCUR: Cantidad de sucursales a recorrer

SUCURSALES: sucursales a las que se debe hacer el recorrido

\$MAX_DINERO: Capacidad máxima que tiene el camión

Distancia_{i,j}: Distancia para ir de sucursal i a sucursal j

VMonto_i: Variación de monto de en sucursal i

Variables

Monto_i: monto con el que cuenta el camión durante el recorrido en el lugar i

Y_{i,j}: bivalente que indica si una sucursal es visitada desde i a j

Restricciones

Llegar una única vez a una sucursal

$$\sum_{\substack{i=0 \\ j \neq i}}^{SUCURSALES} Y_{i,j} = 1 \quad \forall i \in SUCURSALES \cup \text{Central}$$

Salir una única vez a una sucursal

$$\sum_{\substack{j=0 \\ j \neq i}}^{SUCURSALES} Y_{i,j} = 1 \quad \forall j \in SUCURSALES \cup \text{Central}$$

Evitar subtorus

$$U_i - U_j + \text{CANT_SUCUR} * Y_{i,j} \leq \text{CANT_SUCUR} - 1 \quad \forall i, j \in SUCURSALES$$



Restricciones de montos

Nota: por no negatividad de las variables no hace falta validar que $\text{Monto} > 0$.

$$\text{Monto}_i \leq \$\text{MAX_DINERO}$$

$$\text{Monto}_i = \sum_{j=1}^{\text{sucursales}} Y_{i,j} * \text{VMonto}_i$$

Función Objetivo

$$z(\min) = \sum_{i=1}^{\text{SUCURSALES}} \sum_{j=1}^{\text{SUCURSALES}} Y_{i,j} * \text{Distancia}_{i,j}$$