



**DEPARTAMENTO
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

TP2

23 de junio de 2025

Introducción a la Investigación Operativa y Optimización

Integrante	LU	Correo electrónico
Laks, Joaquín	425/22	laksjoaquin@gmail.com
Szabo, Jorge	1683/21	jorgecszabo@gmail.com
Wilders Azara, Santiago	350/19	santiago199913@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón Cero + Infinito)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Conmutador: (+54 11) 5285-9721 / 5285-7400

<https://dc.uba.ar>

1. Modelos

1.1. Modelo para la metodología actual

La metodología actual es idéntico al problema del TSP clásico. Usamos el modelo de Miller, Tucker, y Zemlin para simularlo y como base para las otras metodologías.

1.2. Primer modelo con repartidores

Basado en el modelo anterior, agregamos variables para distinguir cuándo un cliente fue visitado por repartidor a bicicleta y cuándo fue visitado por el camión.

Dado un cliente i definimos D_i como los clientes a distancia menor a $dist_max$ de i .

$$x_{ij} = \begin{cases} 1 & \text{si desde el cliente } v_i \text{ el camión se mueve al cliente } v_j \\ 0 & \text{c.c.} \end{cases}$$

u_i = posición del cliente i en el circuito del camión (no importa cuando el camión no pasa)

$$b_{ij} = \begin{cases} 1 & \text{si se envió un repartidor en bicicleta desde } v_i \text{ hasta } v_j \\ 0 & \text{c.c} \end{cases}$$

b_{ij} solo está definida para $j \in D_i$, por claridad en la formulación está escrito como si estuviera definido para todas las parejas pero se pueden interpretar b_{ij} inválidos como constantes 0.

También contamos con el dato de entrada:

$$r_i = \begin{cases} 1 & \text{si al cliente } v_i \text{ se le entrega un producto que necesita refrigeración} \\ 0 & \text{c.c} \end{cases}$$

Con $n = cant_clientes$, buscamos:

$$\text{Min} \quad \sum_{\substack{v_i, v_j \in V \\ i \neq j}} c_{ij}x_{ij} + costo_repartidor b_{ij}$$

s.a.

$$\sum_{j \neq i} x_{ji} + \sum_{j \neq i} b_{ji} = 1 \quad \forall v_i \in V \quad \text{a toda ciudad se entra una vez, por camión o bicicleta}$$

$$\sum_{j \neq i} x_{ij} = \sum_{j \neq i} x_{ji} \quad \forall v_i \in V \quad \text{si se entró en camión, se sale por camión}$$

$$M(1 - \sum_{j \neq i} b_{ji}) \geq \sum_{j \neq i} b_{ij} + \sum_{j \neq i} x_{ij} \quad \forall v_i \in V, M \geq |V| \quad \text{si se entra en bicicleta, no se sale de ninguna forma}$$

$$\sum_{j \in D_i} b_{ij}r_j \leq 1 \quad \forall v_i \in V \quad \text{ningún repartidor tiene más de un refrigerado}$$

$$u_i - u_j + (n - 1)x_{ij} \leq n - 2 \quad \forall v_i \neq v_j \in V - \{v_1\} \quad \text{continuidad}$$

$$u_1 = 0, 1 \leq u_i \leq n - 1 \quad \forall v_i \in V - \{v_1\}$$

$$x_{ij}, b_{ij} \in \{0, 1\}, u_i \in \mathbb{Z}_{\geq 0}$$

1.3. Modelo con restricciones agregadas

Este modelo extiende el anterior con las siguientes restricciones adicionales:

- Si se contrata un repartidor en una parada de camión determinada, este debe realizar al menos cuatro entregas.
- Hay un conjunto de clientes E que deben ser visitados exclusivamente por un camión. Es decir que sus paquetes no pueden ser repartidos por un repartidor en bicicleta.

Se extiende el modelo presentado en la sección anterior con las siguientes restricciones.

$4(1 - b_{ij}) + \sum_{v_k \in D_i} b_{ik} \geq 4 \quad \forall v_i, v_j \in V \times V$ Si se contrata un repartidor, este pasa por al menos 4 clientes

$\sum_{v_i \in V} b_{ij} = 0 \quad \forall v_j \in E$ Los clientes exclusivos solo son visitados por camión

No hay cambios en las variables, función objetivo o las demás restricciones.

2. Experimentación

Usamos un generador de instancias aleatorias para generar mapas donde algunos caminos son posibles y otros no, asignando distancias y costos al azar, así como también qué clientes necesitan refrigeración o tienen que visitarse con camión en el último modelo. También sacamos datos de posiciones de ciudades de <https://people.sc.fsu.edu/~jburkardt/datasets/cities/cities.html>, usamos las distancias y las modificamos aleatoriamente para crear una instancia con costos.