DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico

17 de julio de 2025 | Intro a IO y Optimización

Integrante	LU	Correo electrónico
Bakal, Ariel	1014/22	bakalariel2002@gmail.com
Bronfman, Dino	868/22	dinobronfman@gmail.com
Stabile, Delfina	819/22	delfistabile18@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

La resolución del trabajo consiste en:
1) La realización de un modelo que calcule el costo de la metodología actual y otro modelo que calcule el costo de la nueva metodología propuesta.
2) Implementar los modelos mediante el paquete CPLEX.
3) Generar instancias que consideren adecuadas para poder sacar conclusiones sobre la experimentación.
Experimentar con las instancias para:
a) Evaluar la ganancia de implemenatar la nueva metodología.
b) Evaluar la pérdida de considerar las restricciones deseables.
c) Analizar los tiempos de cómputo requeridos utilizando diferentes alternativas algorítmicas mediante los parámetros que provee el CPLEX.

1. Modelos

Nosotros utilizamos la formulación de Miller, Tucker and Zemlin para modelar el problema. Siguiendo la metodología actual, el modelo es justamente MTZ sin ningún cambio:

Para cada par de ciudades se definen:

$$x_{ij} = \begin{cases} 1 & \text{si desde la ciudad } v_i \text{ se mueve a la ciudad } v_j, \\ 0 & \text{caso contrario.} \end{cases}$$

Para cada ciudad se define:

$$u_i = \text{posición de la ciudad } v_i \text{ en el circuito.}$$

Y la función junto a las restricciones queda:

$$\begin{array}{lll}
\text{Min} & \displaystyle\sum_{\substack{v_i, v_j \in V \\ j \neq i}} c_{ij} x_{ij} & \text{(costo total de los arcos utilizados)} \\
\text{sujeto a} & \displaystyle\sum_{\substack{v_j \in V \\ j \neq i}} x_{ij} = 1 & \forall v_i \in V \quad \text{(de toda ciudad se debe salir)} \\
& \displaystyle\sum_{\substack{v_j \in V \\ j \neq i}} x_{ji} = 1 & \forall v_i \in V \quad \text{(a cada ciudad se debe llegar)} \\
& u_i - u_j + (n-1)x_{ij} \leq n-2 & \forall v_i, v_j \in V \setminus \{v_1\},\ i \neq j \quad \text{(continuidad)} \\
& u_1 = 0, \quad 1 \leq u_i \leq n-1 & \forall v_i \in V \setminus \{v_1\} \\
& x_{ij} \in \{0,1\}, \quad u_i \in Z & \forall v_i, v_j \in V,\ j \neq i
\end{array}$$

Por otro lado, para la nueva metodología nosotros decidimos adaptar la formulación MTZ y adecuarla según fuera conveniente.
Nombramos:

R**:** Costo de contratar un repartidor por cada cliente que deba visitar = costo repartidor.

D : Distancia máxima entre un cliente y la parada del camión que puede haber para que el producto sea entregado por un repartidor = dist max.

W : Conjunto de clientes con productos a entregar.

W_R : Conjunto de clientes con un pedido refrigerado.

W_E : Conjunto de clientes con pedidos exclusivos.

d_{ij} : distancia entre i y j.

c_{ij} : costo de desplazar el camión del i al j.

A_i : Nodos $j \neq i$ tales que $d_{ij} < D$.

Así es que propusimos las siguientes variables binarias:

$$\delta_i = \begin{cases} 1 & \text{si } i \text{ es parada del camión} \\ 0 & \text{caso contrario.} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{si el camión va de } i \text{ a } j \\ 0 & \text{caso contrario.} \end{cases}$$

$$y_{ki} = \begin{cases} 1 & \text{si } k \text{ es entregada a pie desde } i \\ 0 & \text{caso contrario.} \end{cases}$$

$$u_i = \text{La posición del cliente } i \text{ en el circuito}$$

La propuesta es preprocesar A_i para poder saber a qué clientes se puede alcanzar desde cada parada posible del camión. Es decir, para cada cliente saber a qué otros clientes se puede llegar con un repartidor.

Asumimos que si es optimo entregar a un cliente a pie, simplemente aseguramos que un cliente k fue entregado desde el cliente i con y_{ki}, sin importar como se le entrega. Es decir, no nos interesa saber como el repartidor o los repartidores entregan a esos clientes, ya que, el costo repartidor es solo por cada cliente entregado a pie. Luego para la primera restricción deseable, si vamos a necesitar definir una variable que represente a este repartidor, y obligarlo a entregar a almenos 4 clientes.

De esta manera se resuelve el TSP modificado, que deja de considerar a los clientes que conviene repartir a pie y encuentra un recorrido mínimo del camión entre los clientes que sí deben ser visitados por este.
Así, la función objetivo y las restricciones nos quedan:

$$
\begin{aligned}
Min \quad & \sum_{\substack{i,j \in W \\ j \neq i}} c_{ij} x_{ij} + \sum_{k \in W \setminus \{1\}} \sum_{i \in A_k} R \cdot y_{ki} \\
\text{sujeto a} \quad & (1) \quad \sum_{\substack{k \in A_i \\ k \neq i}} y_{ki} + \sum_{\substack{j \in W \\ j \neq k}} x_{jk} = 1 \\
& (2) \quad y_{ki} \leq \delta_i \quad \forall i \in W, \ \forall k \in A_i \\
& (3) \quad \sum_{j \in W} x_{ij} = \delta_i \quad \forall i \in W \\
& (4) \quad \sum_{j \in W} x_{ji} = \delta_i \quad \forall i \in W \\
& (5) \quad u_i - u_j + (|W| - 1) \cdot x_{ij} \leq (|W| - 2) \quad \forall i, j \in W \setminus \{1\}, i \neq j \\
& \quad u_1 = 0 \\
& \quad 1 \leq u_i \leq 1 + (|W| - 2) \cdot \delta_i \quad \forall i \in W \setminus \{1\} \\
& (6) \quad \delta_1 = 1 \\
& (7) \quad \sum_{k \in W_R} y_{ki} \leq 1 \quad \forall i \in W
\end{aligned}
$$

Además, si quisieramos incluir las "restricciones deseables", necesitamos agregar:
Para la restriccion deseable 1:
Una variable nueva

$$
r_i = \begin{cases} 1 & \text{si se contrata un repartidor desde cliente } i \\ 0 & \text{caso contrario.} \end{cases}
$$

Luego restringimos a contratar al repartidor solo desde una parada del camion y pedimos que si se contrata, entonces haga almenos 4 entregas:

$$
r_i \leq \delta_i \quad \forall i \in W
$$

$$
(|W| - 1) \cdot r_i \geq \sum_{k \in A_i} y_{ki} \geq 4 \cdot r_i \qquad \forall i \in W
$$

Para la restriccion deseable 2:
Obligamos a los clientes exclusivos a ser paradas del camion:

$$
\delta_i = 1 \qquad \forall i \in W_E
$$

Donde:
(1) "Todo cliente es visitado (por el camión o a pie)".
(2) "Los repartidores solo entregan si i es una parada".
(3) "De todo cliente se debe salir".

(4) "A todo cliente se debe llegar".
(5) "Rompimiento de subtour".
(6) "El depósito cuenta como parada del camión".
(7) "Hasta un refrigerado por reparto".

(3) y **(4)** son adaptaciones del TSP ($\sum_j x_{ij} = 1, \sum_j x_{ji} = 1 \quad \forall i$) reemplazando el 1 por δ_i, así se puede modelar que el camión debe llegar y salir de todos los clientes por los que para.

(5) Como las restricciones (3) y (4) me garantizan que $x_{ij} = 1$ si y solo si i y j son paradas ($\delta_i = \delta_j = 1$), tenemos los siguientes casos:
(A) Ambos nodos estan en la ruta del camión ($\delta_i = \delta_j = 1$).
(B) Solo uno esta en la ruta del camión ($\delta_i + \delta_j = 1$).
(C) Ninguno esta en la ruta del camión ($\delta_i + \delta_j = 0$).
En (B) y (C) tenemos $x_{ij} = 0$ por lo anterior (el camion solo se mueve entre nodos que son paradas), entonces la restricción (5) se convierte en $u_i \leq (|W| - 2) + u_j$, la cual es redundante porque las u_i toman valores en el rango $(1, |W| - 1)$.
En (A) tenemos $x_{ij} = 1$, entonces la restriccion (5) me asegura la continuidad de los nodos que son paradas.
Por como modelamos, podriamos utilizar la misma restricción que el modelo de MTZ, ya que la "activamos" unicamente entre nodos que son paradas y para el resto de casos es redundante. Podemos esperar que una solucion optima tome u_i en el rango $(1, |W| - 1)$ cuando $\delta_i = 1$, definiendo asi el orden en que las paradas fueron visitadas y los u_i con $\delta_i = 0$ con valores libres (en nuestro caso CPLEX nos devolvio todos esos $u_i = 1$). Es así que propusimos modificar la restricción que define el rango de u_i como $1 \leq u_i \leq 1 + (|W| - 2) \cdot \delta_i$. Si el nodo i es parada, entonces $1 \leq u_i \leq (|W| - 1)$ y si el nodo i no es parada, entonces $1 \leq u_i \leq 1$, fijandolo en $u_i = 1$ de tal manera de "apagarlos".

2. Instancias

Para comparar los modelos propusimos las siguientes instancias, tratando de cubrir los casos borde, casos particulares y casos generales. Comparamos los 4 modelos: **El TSP original, el TSP modificado (sin las últimas dos condiciones "deseables", el TSP teniendo en cuenta la primera condición deseable y el TSP teniendo en cuenta la segunda).**
Para el TSP modificado analizamos la diferencia en los resultados teniendo en cuenta distintass configuraciones. Estas eran:

Default : Correr CPLEX sin ninguna configuracion.

MIP start : Utilizamos la solucion del TSP Original como solucion inicial del TSP modificado.

tune problem() : Dejamos que CPLEX configure los parametros que considere. prob.parameters.tune problem()

heuristicEffort = 2 : Decide el esfuerzo que toma CPLEX para las heurísticas. El valor default es 1, y nosotros lo evaluamos en 2 ya que es el modo más agresivo, para poder ver bien las diferencias con los otros parámetros.

Gomory Cuts = 2 : Este parámetro controla si se debe hacer cortes de Gomory o no. Seteado en 0 (el default), CPLEX decide si hacer o no, está en automático. En cambio, seteado en 2, se le indica que debe actuar en su modo más agresivo (al igual que el parámetro anterior).

Emphasis = 2 : Este controla el esfuerzo e importancia que le da a la velocidad, viabilidad y optimalidad. Fijado en 2, se le indica enfocarse en la optimalidad sobre la viabilidad.

Además, agregamos un nodo con indice 0 en las instancias. Este vendria a ser el deposito, del cual salimos y tenemos que volver. Para asegurar aquello, definimos distancias de 1 millon y costo 0 a todos los clientes.

Instancia 1: Única parada, conviene repartir todo a pie.

Corrimos una instancia con:

- 6 clientes
- Costo por cada entrega a pie o en bicicleta: 100
- Distancia máxima recorrible por un repartidor: 5
- Entregas refrigeradas: 0
- Pedidos exclusivos: 1

Se espera que el modelo opte por una única parada del camión y que las entregas se realicen mayormente a pie, aprovechando la proximidad de los clientes para reducir costos.

Obtuvimos los siguientes resultados:

Modelo	Resultado de la función objetivo
TSP original	4000 (16 ms)
TSP modificado	400 (< 1 ms)
TSP con condición deseable 1	400 (15 ms)
TSP con condición deseable 2	1300 (< 1 ms)

Notemos cómo utilizar nuestros modelos modificados mejoran la solucion, ya que en este caso, es mas barato hacer una parada y entregar todo a pie. Notemos ademas que el TSP modificado y el TSP con condicion deseable 1 tienen el mismo resultado, esto es porque la mejor solucion es entregar a mas de 4 clientes desde una parada (que es lo que exige la restriccion). Por último, con el TSP con condicion deseable 2, era esperado obtener un resultado mejor que el TSP original pero peor que los demas TSP modificados, ya que, exijimos hacer una parada en un nodo que o no alcanza a todos los clientes a pie o bien es caro llegar allí con el camión.

Comparando los distintos parámetros tomados con el TSP modificado, notamos que la mayoría tuvo un tiempo de ejecución similar, salvo el tune problem que tuvo uno de 46 ms y el de emphasis que tuvo el mejor, con menos de 1 ms. Esto que ocurrió con el tune problem era de esperarse, ya que CPLEX se pone un tiempo a configurar parametros antes de intentar encontrar una solución. Es así que en todas las instancias donde tomamos este parámetro el desempeño empeoró.

Instancia 2: Todos exclusivos (camión obligatorio).

Corrimos una instancia con:

- 6 clientes
- Costo por cada entrega a pie o en bicicleta: 100
- Distancia máxima recorrible por un repartidor: 5
- Entregas refrigeradas: 0
- Pedidos exclusivos: 5

Se espera que el camión realice todas las entregas porque todos los clientes fueron definidos como exclusivos, lo que obliga a que sean visitados por el camión, independientemente de costos o distancias.

Modelo	**Resultado del objetivo**
TSP original	4000 (< 1 ms)
TSP modificado	400 (< 1 ms)
TSP con condición deseable 1	400 (16 ms)
TSP con condición deseable 2	4000 (47 ms)

Esta instancia sirve para apreciar cómo en casos donde hay muchos pedidos exclusivos el último modelo tiene un peor desempeño que los otros. Esto es porque se ve forzado a hacer muchas más paradas (en este caso se ve forzado a hacer todas las paradas ya que son todos exclusivos). Por como están ubicadas las casas, no empeora el resultado al agregar la restricción deseable 1 (esto es circunstancial). Comparando los distintos parámetros tomados notamos que todos tuvieron un desempeño muy similar (a excepción del tune problem), rondeando los 16 ms. Por eso en este caso particular la configuración default superó a estos otros, ya que tuvo un tiempo de < 1 ms.

Instancia 3: Un único cliente aislado se atiende en camión, el resto a pie.

Corrimos una instancia con:

- 51 clientes
- Costo por cada entrega a pie o en bicicleta: 5
- Distancia máxima recorrible por un repartidor: 2
- Entregas refrigeradas: 0
- Pedidos exclusios: 0

Se espera que el modelo asigne una entrega directa en camión para el cliente aislado por su ubicación o costo, y que para el resto se prefiera la entrega a pie desde algunas paradas del camión.
En esta instancia no variaron en general los resultados del TSP modificado con los distintos parámetros tomados:

Configuraciones	**Resultados**
Default	439 (TLE)
tune problem()	434 (TLE)
MIP start	434 (TLE)
heuristic Effort	439 (TLE)
Gomory cuts	434 (TLE)
Emphasis	434 (TLE)

Sí cabe destacar que todos (incluso, cómo se muestra a continuación, los otros modelos) tuvieron un TLE (time limit excedeed) y estos son los valores a los que llegaron. En este caso el default y heuristic Effort tuvieron un desempeño ligeramente peor.

Modelo	**Resultado de la función objetivo**
TSP original	613 (TLE)
Mejor resultado del TSP modificado	434 (TLE)
TSP con condición deseable 1	462 (TLE)
TSP con condición deseable 2	434 (TLE)

Como se puede ver, en este caso el TSP original tuvo un peor desempeño que los otros. Esto tiene sentido porque la instancia propuso una situación donde convenía entregar los pedidos mediante repartidores en vez de ir trasladando el camión de parada en parada. Al mismo tiempo, dentro de los 3 modelos del TSP modificado, como sucedio en casi todas las instancias, el modelo con la primera restricción deseada tuvo un peor desempeño, ya que este suele forzar más paradas del camión. En esta ocasión el modelo con la segunda restricción deseable tuvo un desempeño casi igual al que no contempla estas últimas restricciones, pero esto depende mucho de cuántos pedidos exclusivos había.

Instancia 4: 2 Refrigerados en cada cluster.

Corrimos una instancia con:

- 6 clientes
- Costo por cada entrega a pie o en bicicleta: 100
- Distancia máxima recorrible por un repartidor: 5
- Entregas refrigeradas: 0
- Pedidos exclusivos: 1

Algunos clientes refrigerados dispersos obligan al camión a hacer varias paradas. El uso de repartidores es limitado, por lo que el ahorro es moderado. En esta instancia hay 2 pedidos refrigerados y esto rompe los clusters, ya que obligan al camión a hacer más paradas.

En esta instancia, si bien no varió el valor final, si cambió mucho el tiempo de ejecución de cada parámetro. En este ocasión, heuristic Effort y Gomory Cuts tuvieron un mejor desempeño, mientras que (como sucedio en general) tomando el parámetro de tune problem el tiempo empeoró considerablemente.

Configuraciones	**Resultados**
Default	1004700 (156 ms)
tune problem()	1004700 (515 ms)
MIP start	1004700 (187 ms)
heuristic Effort	1004700 (94 ms)
Gomory cuts	1004700 (94 ms)
Emphasis	1004700 (109 ms)

Tomando en cuenta el mejor valor posible del TSP modificado, la tabla de valores nos quedó:

Modelo	**Resultado de la función objetivo**
TSP original	6006000 (297 ms)
TSP modificado	1004700 (94 ms)
TSP con condición deseable 1	6006000 (78 ms)
TSP con condición deseable 2	1004700 (93 ms)

Efectivamente podemos observar lo esperado, romper por tener 2 refrigerados a los clusters obliga a hacer mas paradas. En el deseable 1 obtuvimos el mismo resultado que el TSP Original ya que no tenemos mas clusters que se puedan entregar a almenos 4 clientes.
Curiosamente, en esta instancia el tiempo de ejecución del TSP con la primera condición deseable fue (si bien ligeramente) mejor que el del TSP modificado. Esto solo ocurrió en esta instancia.

Instancia 5: Clusteres separados.

Corrimos una instancia con:

- 14 clientes

- Costo por cada entrega a pie o en bicicleta: 100
- Distancia máxima recorrible por un repartidor: 5
- Entregas refrigeradas: 0
- Pedidos exclusivos: 2

Clientes agrupados en clústeres separados espacialmente. El camión visita cada clúster y repartidores hacen entregas locales, lo que reduce costos significativamente. La instancia formulada tiene 3 clusters, y 2 de estos tienen solamente 3 clientes (sin contar la parada del camión) lo suficientemente cerca para repartirles los pedidos a pie.

Modelo	Resultado de la función objetivo
TSP original	6006000 (47 ms)
TSP modificado	3000 (31 ms)
TSP con condición deseable 1	4004400 (47 ms)
TSP con condición deseable 2	3900 (47 ms)

En este caso el único parámetro que empeoró el tiempo de ejecución del TSP modificado fue el de tune problem, que dio 141 ms.En la tabla nos quedamos con el mejor valor dado, que fue el alcanzado con el resto de los parámetros
Los resultados fueron los esperados. El TSP original obtuvo un resultado altísimo, ya que tuvo que recorrer grandes distancias. En cambio, el TSP modificado pudo repartir todos estos a pie. Además, al haber solo 1 cluster con 4 clientes (sin contar la parada) a distancia repartible a pie, el TSP con la condición deseable 1 no pudo aprovechar mucho los repartidores. Además, al haber un cliente exclusivo el modelo con la segunda condición deseable tuvo que hacer una parada extra.

Instancia 6: Solo 3 clientes a pie, y 1 solo exclusivo. El resto con el camión.

Corrimos una instancia con:

- 5 clientes
- Costo por cada entrega a pie o en bicicleta: 100
- Distancia máxima recorrible por un repartidor: 5
- Entregas refrigeradas: 0
- Pedidos exclusivos: 1

Modelo	Resultado de la función objetivo
TSP original	3000 (16 ms)
TSP modificado	300 ($<$ 1 ms)
TSP con condición deseable 1	3000 (16 ms)
TSP con condición deseable 2	1200 (16 ms)

En esta última instancia, el TSP modificado obtuvo un mejor tiempo de ejecución cuando utilizamos el parámetro de Gomory Cuts (< 1 ms), pero todos los otros tardaron 16 ms, salvo tune problem que tardó 47 ms.
En este caso podemos observar cómo se ve afectado el TSP con la condición deseable 1 cuando no hay 4 paradas para repartir a pie juntas. Al haber solo 3, se ve obligado a hacer todas las paradas con el camión y así seguir la metodología original, llegando al mismo resultado que el TSP original. En cambio, los otros dos modelos sí pudieron repartir a pie esos 3, y es por eso que obtuvieron un resultado mucho mejor. Sin embargo, al haber uno exclusivo el TSP con la segunda condición deseable tuvo que hacer una parada extra con el camión, y eso se vio reflejado en su resultado.

Conclusión

Teniendo en cuenta estas instancias podemos apreciar que si bien los resultados fueron variando, el modelo del TSP modificado (sin las restricciones deseables) fue el mejor. Este siempre obtuvo un valor mejor o igual que los otros. Es decir, el peor caso siempre va a ser el del TSP original ya que es el que menos herramientas tiene para repartir los pedidos: siempre debe entregar con el camión. Los otros tres modelos también pueden optar por esta opción, pero también tienen la posibilidad de encontrar mejores resultados utilizando los repartidores. La instancia donde más se puede apreciar la diferencia entre la metodología original y la nueva es en la 5, de los clusters. En esta instancia la diferencia entre los costos es abismal, ya que siguiendo la nueva metodología se aprovecha por completo el uso de los repartidores, mientras que en la original el camión tiene que recorrer todas las paradas.

Además, analizando los distintos tiempos de ejecución, se puede observar que en general, con la salvedad de la instancia 4, el TSP modificado siempre fue mejor o igual que el resto en este aspecto.

Es por esto que podemos afirmar que conviene utilizar la nueva metodología. Al mismo tiempo, al estar menos restringido, el TSP modificado sin las restricciones deseables fue el que mejor desempeño tuvo. Por otro lado, evaluando los otros dos modelos que contemplan las distintas restricciones deseables, notamos:

Modelos TSP con restricciónes deseables 1 o 2: El primero de estos (deseable 1) tuvo en general un peor desempeño que el que tiene en cuenta la otra restricción. Esto es porque limita mucho el uso de los repartidores. Por ejemplo, viendo la instancia 6 se puede apreciar facilmente que cuando no hay 4 pedidos juntos que se puedan repartir a pie (pero sí hay alguno), este tiene un desempeño igual al del TSP original.
El único caso en el que este modelo supera al otro es en el que la cantidad de pedidos exclusivos sea mayoritaria, porque limita incluso más el uso de los repartidores. Esto se puede apreciar en la segunda instancia evaluada, donde los pedidos son todos exclusivos, y siguiendo este último modelo el camión se vio obligado a hacer cada parada. De esta manera, mientras que el TSP modificado y el TSP que contempla la restricción deseable 1 tuvieron un buen desempeño, los otros dos tuvieron un resultado mucho peor. Sin embargo, en general, si no hay muchos pedidos exclusivos,

este modelo (entre estos dos) es el que menos limita las opciones para repartir.
En cuanto a su velocidad, no se pueden sacar conclusiones determinísticas entre estos dos modelos ya que sus tiempos de ejecución fueron variando (si bien no mucho) dependiendo la instancia pero no estuvieron relacionados con sus valores. Es decir, no distinguimos como algo generalizable que el que diera mejor valor tardaba más o menos.

Analizando los resultados obtenidos tomando diferentes parámetros para el TSP modificado notamos que salvo en la tercera instancia (donde todos se vieron interrumpidos por el TLE) ninguno afectó el valor final, y en general todos salvo 'tune problem' tuvieron un desempeño parecido en cuanto al tiempo de ejecución.