

Optimización para la Toma de Decisiones - Examen final

Coronel, Tomás y Villarinos, Damián

Diciembre 2025

1. Primera parte

1.1. Introducción

En este trabajo se estudia un problema de ruteo con múltiples tipos de vehículos y un conjunto amplio de restricciones operativas. El objetivo es analizar el impacto tanto del **modelado matemático** como de la **parametrización del solver CPLEX** sobre el desempeño computacional y la calidad de las soluciones.

Se comparan dos enfoques:

- **Método Actual:** formulación base, con un conjunto reducido de restricciones.
- **Método Nuevo:** formulación extendida que incorpora restricciones adicionales tales como exclusividad de clientes, precedencias, límites por tipo de vehículo y balance modal.

1.2. Metodología Experimental

Los experimentos fueron ejecutados utilizando CPLEX bajo las siguientes condiciones comunes:

- Un único hilo de ejecución (`threads = 1`).
- Gap de optimalidad del 0.5 %.
- Registro completo del log del solver para cada ejecución.

Las ejecuciones se organizaron en 14 escenarios, cada uno correspondiente a una parametrización específica de CPLEX (Tabla A), evaluada sobre ambos métodos. Como resultado, se realizaron un total de 28 ejecuciones. Cada ejecución se identifica mediante un nombre que explica tanto el modelo utilizado como los parámetros relevantes del solver: estrategia de selección de nodos, frecuencia de heurísticas, nivel de cortes y uso de presolve.

1.3. Infeasibilidad del Modelo

Durante la experimentación con el **Método Nuevo** se observaron instancias infeasibles. Este comportamiento no se debe a errores de implementación, sino a la interacción estructural entre restricciones.

En particular, la combinación de:

- clientes exclusivos,
- restricciones sobre clientes refrigerados,
- precedencias obligatorias,

- y límites mínimos de asignación por tipo de vehículo,

reduce significativamente el espacio factible. La detección de infeasibilidad por parte del solver constituye un resultado relevante, ya que evidencia configuraciones operativas inviables desde el punto de vista del negocio.

1.4. Resultados

Los resultados detallados de cada ejecución se presentan en el Anexo (Tabla B), donde se reportan el estado final, el valor de la función objetivo, el tiempo de cómputo y observaciones cualitativas sobre el comportamiento del solver.

1.4.1. Método Actual

El método actual exhibe un comportamiento estable y consistente a lo largo de todas las configuraciones evaluadas, alcanzando siempre el mismo valor óptimo de la función objetivo. Las diferencias observadas entre escenarios se manifiestan principalmente en el tiempo de cómputo y en la dinámica de exploración del árbol de Branch-and-Bound.

Las configuraciones con heurísticas agresivas permiten obtener incumbentes tempranos, lo que reduce la necesidad de explorar un gran número de nodos. Sin embargo, en ausencia de cortes, la mejora del bound dual es progresiva, lo que explica la persistencia de gaps elevados en etapas iniciales del proceso.

Las configuraciones conservadoras, caracterizadas por la desactivación de heurísticas y cortes, tienden a realizar una exploración más sistemática del árbol, resultando en tiempos de resolución relativamente bajos debido al tamaño reducido del modelo, aunque con soluciones iniciales de menor calidad.

La incorporación de cortes, especialmente en niveles agresivos, fortalece la formulación y mejora significativamente los bounds duales, limitando el branching necesario. No obstante, en este método, dicha mejora no siempre se traduce en una reducción sustancial del tiempo total. Finalmente, se observa que la desactivación del presolve incrementa el tamaño efectivo del modelo y la cantidad de nodos explorados, afectando negativamente el desempeño general.

1.4.2. Método Nuevo

El método nuevo introduce un conjunto ampliado de restricciones que incrementa considerablemente la cantidad de variables binarias y restricciones del modelo. Este aumento de complejidad tiene un impacto directo en el desempeño del solver, manifestándose en tiempos de cómputo sensiblemente mayores y en una exploración más extensa del árbol de Branch-and-Bound.

En este contexto, las heurísticas primales juegan un rol fundamental, ya que permiten encontrar soluciones factibles en etapas tempranas del proceso. No obstante, aun contando con incumbentes tempranos, el cierre del gap requiere un esfuerzo computacional significativo, reflejando la dificultad estructural del problema extendido.

El uso de cortes resulta particularmente efectivo para mejorar el bound dual y reducir el branching. Las configuraciones con cortes agresivos logran acelerar la convergencia en varios escenarios, aunque a costa de un mayor tiempo de procesamiento por nodo. Asimismo, el presolve se revela como un componente crítico: su desactivación conduce a un crecimiento sustancial del árbol y a tiempos de resolución considerablemente más elevados, aun cuando se emplean heurísticas y cortes.

1.4.3. Comparación Global

A partir del análisis comparativo se desprenden las siguientes conclusiones generales:

- El método actual resulta computacionalmente más sencillo y robusto, aunque su capacidad de representación del problema real es limitada.
- El método nuevo modela de forma más fiel la realidad operativa, a costa de un incremento significativo en la complejidad computacional y en los tiempos de resolución.
- No existe una parametrización de CPLEX que sea universalmente dominante: el desempeño observado depende fuertemente de la interacción entre la formulación del modelo y los parámetros del solver, en particular heurísticas, cortes y presolve.

En ambos métodos se observa que el valor óptimo obtenido es robusto frente a la parametrización del solver, siendo el tiempo de cómputo el principal factor afectado por las distintas configuraciones evaluadas.

1.5. Conclusiones

El trabajo demuestra que la parametrización del solver es un componente clave en problemas de optimización entera mixta. Asimismo, se evidencia que el enriquecimiento del modelo puede llevar a instancias infeasibles, lo cual constituye información valiosa para la toma de decisiones.

La exploración sistemática realizada permite comprender mejor el comportamiento de CPLEX y resalta la importancia de adaptar tanto el modelo como los parámetros al contexto del problema.

Desde un punto de vista práctico, los escenarios que combinan heurísticas moderadas con cortes bien calibrados ofrecen el mejor compromiso entre calidad de solución y tiempo de resolución, especialmente en modelos de mayor complejidad.

2. Segunda parte

1. Definición del modelo

Conjuntos:

- $i \in I = \{1, \dots, 6\}$: Centros de distribución.
- $j \in J = \{1, \dots, 8\}$: Zonas de entrega.

Parámetros:

- f_i : costo fijo de apertura del centro i .
- c_{ij} : costo unitario de transporte desde el centro i a la zona j .
- d_j : demanda de la zona j .
- u_i : capacidad máxima del centro i .
- a_{ij} : parámetro de cobertura.

$$a_{ij} = \begin{cases} 1 & \text{si el centro } i \text{ puede atender a la zona } j, \\ 0 & \text{en caso contrario.} \end{cases}$$

Variables de decisión:

$$y_i = \begin{cases} 1 & \text{si el centro } i \text{ es abierto,} \\ 0 & \text{en caso contrario,} \end{cases} \quad x_{ij} = \begin{cases} 1 & \text{si el centro } i \text{ atiende a la zona } j, \\ 0 & \text{en caso contrario.} \end{cases}$$

Función objetivo

$$\min \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} d_j x_{ij}$$

Restricciones

- $\sum_{i \in I} x_{ij} = 1 \quad \forall j \in J$ Cada zona es atendida por un único centro
- $x_{ij} \leq y_i \quad \forall i \in I, j \in J$ Solo asigno zonas a centros que estén abiertos
- $x_{ij} \leq a_{ij} \quad \forall i \in I, j \in J$ Compatibilidad con la matriz de cobertura
- $\sum_{j \in J} d_j x_{ij} \leq u_i y_i \quad \forall i \in I$ La demanda de cada zona no puede exceder la capacidad del centro
- $\sum_{i \in I} y_i = 3$ Se deben abrir exactamente 3 centros
- $y_2 + y_5 \leq 1$ Si se abre el centro 2, no puede abrirse el centro 5
- $\sum_{j \in J} x_{1j} \geq 2y_1$ Si se abre el centro 1, debe atender al menos 2 zonas
- $y_i \in \{0, 1\} \quad \forall i \in I$
- $x_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J$

2. Preguntas sobre el modelo

- ¿Cuál es la mejor cota LB? y UB?
 - La mejor cota LB (Lower Bound) que encontramos se haya en el Nodo 4, con un z^* entero igual a 94.
 - La mejor cota UB (Upper Bound) que se encuentra en el árbol corresponde a la raíz del problema, con un valor objetivo de la relajación igual a 99.

- b) ¿Qué nodos pueden ser cerrados?
- El nodo 3, dado que es infactible.
 - El nodo 4, al ser una solución entera el nodo se cierra y se toma como posible cota/solución
 - El nodo 7 es inferior a la cota inferior, por lo que se descarta para exploración.
- c) ¿Qué nodo procesaría si usara la regla Mejor cota?
- Si utilizáramos el criterio de *Mejor cota*, elegiríamos procesar el Nodo 6, cuyo valor es de 96 (fraccionario), frente a los nodos 7 y 8 con valores 92 y 95, respectivamente, ambos fraccionarios.
- d) ¿Qué nodo procesaría si usara la regla Profundidad?
- Utilizando la regla de *profundidad*, el siguiente nodo a ser procesado sería el número 8, dado que el número 7 sería podado por ser inferior a la cota inferior.

3. Verdadero / Falso

- a) La única posibilidad de cerrar un nodo del árbol de B&B es si la relajación lineal asociada es no factible.
- Falso.** Un nodo de un árbol de B&B no solo puede cerrarse si la relajación lineal asociada es infactible, sino que también se cierra si el subproblema tiene una solución lineal con un valor de la función objetivo peor que el de la mejor solución.
- b) Si el óptimo de la relajación lineal tiene coordenadas enteras entonces es el óptimo del problema entero.
- Falso.** No necesariamente si el óptimo de la relajación lineal es un entero entonces es el óptimo del problema entero. La solución se convierte en candidata para mejor solución, pero si en otro nodo se encuentra una mejor solución, esta se reemplaza.
- c) En un algoritmo tipo Branch and Cut o tipo Branch and Bound, el gap nos da una cota superior del error de la mejor solución encontrada hasta el momento.
- Verdadero.** El gap nos da un porcentaje de hasta cuánto se podría mejorar la solución actual.
- d) Un recorrido en profundidad del árbol de B&B siempre es más eficiente que un recorrido a lo ancho.
- Falso.** No necesariamente un camino tiene que ser siempre mejor que otro, ambos tienen partes positivas y negativas. Una búsqueda en profundidad puede llevar a callejones sin salida, mientras que una búsqueda a lo ancho puede requerir un gran uso de memoria.
- e) En un problema de programación lineal entera de maximización, el valor óptimo de la relajación lineal provee una cota superior del valor óptimo entero.
- Falso.** En un problema de maximización lineal entera la cota superior es el mayor valor objetivo de la relajaciones de nodos abiertos.
- f) Un algoritmo tipo Branch and Cut puede utilizarse como un algoritmo heurístico.
- Verdadero.** Un algoritmo tipo Branch and Cut funciona dado que extrae cotas superiores obtenidas mediante heurísticas inicial y/o primal.
- g) Un algoritmo de separación no puede ser un algoritmo heurístico.
- Falso.** Los algoritmos de separación pueden ser tanto exactos como heurísticos.
- h) Siempre es mejor realizar el proceso de branching basado en la variable más fraccionaria.
- Falso.** Existen diversas estrategias de ramificación (variable mas fraccionaria, variable menos fraccionaria, variable con mayor importancia económica, por restricciones, por criterios ad-hoc) que pueden resultar mejores dependiendo del tipo de problema a atacar.

- i) Las desigualdades cover pueden aplicarse como planos de corte a cualquier problema de programación lineal entera binaria.
Verdadero. Las desigualdades de cubrimiento son problemas de planos de corte generales como también los cortes de Gomory o los cliques.
- j) Redondear una solución óptima fraccionaria nunca brinda el óptimo entero.
Verdadero. Si bien pueden existir casos en donde redondeando se logre llegar al óptimo entero, en los casos de variables 0-1 esto puede dar un resultado muy lejano al óptimo.
- k) En el caso que un problema sea factible, un algoritmo tipo Branch and Cut siempre encuentra el óptimo.
Verdadero. Debido a su buena formulación y buena resolución de cotas inferiores y superiores, un algoritmo de Branch and Cut que sea factible siempre va a lograr llegar a soluciones óptimas. Solamente fallarían por cuestiones de memoria e imposibilidad de completar todos los nodos.
- l) En el caso que un problema sea factible, un algoritmo de planos de corte siempre converge a la solución óptima.
Falso. Un algoritmo de plano de corte de un problema factible no siempre encuentra la solución óptima. En algunos casos solamente consigue identificar la desigualdad válida para realizar un problema de separación.
- m) Un algoritmo tipo Branch and Cut debe usar una heurística primal.
Falso. Si bien las heurísticas primales son una excelente herramienta para resolver problemas de Branch and Cut, no es obligatorio su uso.
- n) Redondear una solución óptima de la relajación lineal nunca puede dar una solución óptima entera.
Falso. Redondear la solución de la relajación lineal en algunos casos puede dar una solución lineal entera, aunque no es lo más recomendable.
- ñ) Hay problemas de programación lineal entera fáciles de resolver.
Verdadero. Como en todos los procedimientos, siempre hay problemas cuya solución se encuentra más rápido que en otros problemas.
- o) Un algoritmo de separación siempre encuentra una restricción violada (si existe).
Falso. En el caso de utilizar algoritmos de separación heurísticos es posible que existan desigualdades violadas pero que el procedimiento no sea capaz de encontrarlas.

A. Escenarios evaluados y parametrización de CPLEX

Escenario	Parámetros CPLEX
ns1_hfA_cuts0_ps1_baseline	ns=1, hf=-1, c=0, ps=1
ns1_hfA_cuts0_ps0_baseline_sin_ps	ns=1, hf=-1, c=0, ps=0
ns0_hf0_cuts0_ps0_conservadora	ns=0, hf=0, c=0, ps=0
ns0_hf0_cuts0_ps1_conservadora_ps	ns=0, hf=0, c=0, ps=1
ns0_hf1_cuts0_ps1_heur_agresiva	ns=0, hf=1, c=0, ps=1
ns0_hf1_cuts0_ps0_heur_agresiva_sin_ps	ns=0, hf=1, c=0, ps=0
ns2_hf5_cuts0_ps1_heur_moderada	ns=2, hf=5, c=0, ps=1
ns1_hfA_cuts1_ps1_cuts_moderados	ns=1, hf=-1, c=1, ps=1
ns1_hfA_cuts2_ps1_cuts_agresivos	ns=1, hf=-1, c=2, ps=1
ns1_hfA_cuts2_ps0_cuts_agresivos_sin_ps	ns=1, hf=-1, c=2, ps=0
ns2_hf5_cuts1_ps1_mixto_balanceado	ns=2, hf=5, c=1, ps=1
ns1_hf1_cuts2_ps1_mixto_agresivo	ns=1, hf=1, c=2, ps=1
ns0_hf1_cuts1_ps1_rapido	ns=0, hf=1, c=1, ps=1
ns0_hf1_cuts0_ps0_rapido_minimo	ns=0, hf=1, c=0, ps=0

Referencia de parámetros:

- ns: estrategia de selección de nodos (`mip.strategy.nodeselect`).
- hf: frecuencia de heurísticas primales (`mip.strategy.heuristicfreq`).
- c: nivel de cortes aplicado en forma uniforme a `covers`, `flowcovers` y `mircut`.
- ps: activación del presolve (`preprocessing.presolve`).

B. Resultados obtenidos y comportamiento observado del solver

Método	Escenario	Estado	Costo	Tiempo (s)	Observaciones del proceso
Actual	ns1_hfA_cuts0_ps1_baseline	OPT	264.0	18.061	Exploración moderada del árbol; convergencia progresiva sin apoyo de cortes.
Actual	ns1_hfA_cuts0_ps0_baseline_sin_ps	OPT	264.0	20.362	Aumento del tamaño efectivo del modelo; mayor exploración por falta de presolve.
Actual	ns0_hf0_cuts0_ps0_conservadora	OPT	264.0	5.483	Árbol pequeño; resolución rápida debido a baja complejidad del modelo.
Actual	ns0_hf0_cuts0_ps1_conservadora_ps	OPT	264.0	5.737	Presolve reduce el modelo, aunque sin impacto significativo en el tiempo total.
Actual	ns0_hf1_cuts0_ps1_heur_agresiva	OPT	264.0	5.970	Incumbente obtenido tempranamente mediante heurísticas; branching limitado.
Actual	ns2_hf5_cuts0_ps1_heur_moderada	OPT	264.0	10.626	Exploración más equilibrada del árbol; heurísticas menos dominantes.
Actual	ns0_hf1_cuts0_ps0_heur_agresiva_sin_ps	OPT	264.0	5.959	Heurísticas compensan la ausencia de presolve; árbol contenido.
Actual	ns1_hfA_cuts1_ps1_cuts_moderados	OPT	264.0	9.519	Mejora temprana del bound dual; cortes reducen la exploración del árbol.
Actual	ns1_hfA_cuts2_ps1_cuts_agresivos	OPT	264.0	9.476	Cortes efectivos fortalecen el modelo sin sobrecargar el tiempo por nodo.
Actual	ns1_hfA_cuts2_ps0_cuts_agresivos_sin_ps	OPT	264.0	9.350	Cortes compensan parcialmente la falta de presolve; branching controlado.
Actual	ns2_hf5_cuts1_ps1_mixto_balanceado	OPT	264.0	10.523	Buen compromiso entre heurísticas y cortes; árbol de tamaño moderado.
Actual	ns1_hf1_cuts2_ps1_mixto_agresivo	OPT	264.0	15.029	Configuración agresiva aumenta el costo por nodo sin mejoras sustanciales.
Actual	ns0_hf1_cuts1_ps1_rapido	OPT	264.0	6.607	Estrategia orientada a velocidad; incumbente temprano y cierre rápido.
Actual	ns0_hf1_cuts0_ps0_rapido_minimo	OPT	264.0	6.466	Configuración mínima; árbol reducido y solución estable.
Nuevo	ns1_hfA_cuts0_ps1_baseline	OPT	232.0	56.212	Modelo más complejo; árbol grande y convergencia lenta del gap.
Nuevo	ns1_hfA_cuts0_ps0_baseline_sin_ps	OPT	232.0	90.201	Sin presolve el árbol crece significativamente; branching extensivo.
Nuevo	ns0_hf0_cuts0_ps0_conservadora	OPT	232.0	36.292	Exploración profunda del árbol; heurísticas desactivadas enlentecen la convergencia.
Nuevo	ns0_hf0_cuts0_ps1_conservadora_ps	OPT	232.0	79.751	Presolve insuficiente para compensar la falta de heurísticas.
Nuevo	ns0_hf1_cuts0_ps1_heur_agresiva	OPT	232.0	84.624	Incumbente temprano pero alto costo computacional en el cierre del gap.
Nuevo	ns2_hf5_cuts0_ps1_heur_moderada	OPT	232.0	14.333	Buen equilibrio entre heurísticas y branching; árbol controlado.
Nuevo	ns0_hf1_cuts0_ps0_heur_agresiva_sin_ps	OPT	232.0	44.970	Falta de presolve incrementa el tamaño del árbol pese a heurísticas activas.
Nuevo	ns1_hfA_cuts1_ps1_cuts_moderados	OPT	233.0	11.930	Cortes mejoran el bound dual; reducción efectiva del branching.
Nuevo	ns1_hfA_cuts2_ps1_cuts_agresivos	OPT	232.0	11.171	Cortes altamente efectivos; rápida convergencia con árbol reducido.
Nuevo	ns1_hfA_cuts2_ps0_cuts_agresivos_sin_ps	OPT	232.0	72.444	Cortes no logran compensar la ausencia de presolve; alto costo por nodo.
Nuevo	ns2_hf5_cuts1_ps1_mixto_balanceado	OPT	232.0	17.866	Compromiso razonable entre robustez y tiempo computacional.
Nuevo	ns1_hf1_cuts2_ps1_mixto_agresivo	OPT	232.0	16.196	Configuración agresiva con buen balance entre cortes y heurísticas.
Nuevo	ns0_hf1_cuts1_ps1_rapido	OPT	232.0	9.743	Mejor desempeño temporal del método nuevo; árbol pequeño.
Nuevo	ns0_hf1_cuts0_ps0_rapido_minimo	OPT	232.0	52.618	Configuración mínima sin presolve; exploración extensa del árbol.