



Desafío de Localización y Ruteo

SMIO–Hexaly

Monterrey 2026

Especificación del problema

Versión 1.0

Preparado por: Eduardo Salazar
Vocal, SMIO

En representación del Consejo Directivo de la SMIO

Abril de 2026

Índice

1. Descripción del problema	2
1.1. El Problema de Localización y Ruteo con Capacidades (CLRP)	2
1.2. Relevancia industrial	2
2. Datos de entrada	2
2.1. Conjuntos	2
2.2. Parámetros de los depósitos	3
2.3. Parámetros de los clientes	3
2.4. Parámetros de vehículos y ruteo	3
2.5. Distancias	3
3. Decisiones y restricciones	3
3.1. Decisiones	3
3.2. Restricciones	4
3.3. Función objetivo	4
4. Formatos de archivo	4
4.1. Formato de instancia	5
4.2. Formato de solución	5
4.3. Verificación de soluciones	6
5. Diseño de instancias	6
5.1. Variación estructural	6
6. Formato de la competencia	7
6.1. Categorías	7
6.2. Puntuación y tabla de clasificación	7
6.2.1. Solución base de Hexaly y BKS de los participantes	7
6.2.2. Puntuación por tiempo de liderazgo	8
6.2.3. Tablas de clasificación	8
6.3. Descripción del método	9

1. Descripción del problema

1.1. El Problema de Localización y Ruteo con Capacidades (CLRP)

Consideramos el **Problema de Localización y Ruteo con Capacidades (CLRP, por sus siglas en inglés)**, una extensión del problema clásico que combina decisiones estratégicas de localización de instalaciones con decisiones operativas de ruteo de vehículos.

Se proporcionan un conjunto de ubicaciones potenciales de depósitos y un conjunto de clientes con demandas conocidas. Se dispone de una flota de vehículos idénticos para atender a los clientes mediante rutas que inician y terminan en un depósito abierto. El objetivo es decidir simultáneamente qué depósitos abrir, asignar los clientes a los depósitos abiertos y diseñar las rutas de los vehículos, minimizando el costo total.

Esta formulación extiende el CLRP clásico con dos enriquecimientos operativos: un costo fijo que se incurre cada vez que se despacha un vehículo y un número máximo de vehículos que cada depósito puede albergar.

El sitio web del desafío incluirá una instancia de ejemplo, un formato de solución factible y el código del validador de soluciones, de modo que los participantes puedan comprender qué constituye una solución factible y cómo se calcula la función objetivo.

1.2. Relevancia industrial

El CLRP es directamente relevante para la industria mexicana. Empresas como OXXO, Bimbo y FEMSA enfrentan de manera rutinaria decisiones sobre dónde ubicar sus centros de distribución mientras optimizan simultáneamente las rutas de entrega. Plantear el desafío en torno a este problema resuena tanto con audiencias académicas como industriales.

Desde el punto de vista algorítmico, el LRP se sitúa en un punto óptimo de dificultad: combina decisiones estratégicas de localización con ruteo operativo, lo que implica que ningún paradigma algorítmico domina por sí solo. Esto abre la competencia a equipos que empleen metaheurísticas, métodos de descomposición, branch-and-price, matheurísticas, aprendizaje automático o enfoques híbridos.

Además, existe una menor saturación de instancias de referencia en comparación con el CVRP, por lo que el desafío llena un vacío genuino en la comunidad y cualquier resultado conlleva una mayor novedad.

2. Datos de entrada

2.1. Conjuntos

- $I = \{1, \dots, m\}$: conjunto de ubicaciones potenciales de depósitos.
- $J = \{1, \dots, n\}$: conjunto de clientes.

2.2. Parámetros de los depósitos

Para cada depósito $i \in I$:

- (x_i^d, y_i^d) : coordenadas del depósito i .
- f_i : costo fijo de abrir el depósito i .
- W_i : capacidad del depósito i (demanda total máxima que puede atender).
- V_i : número máximo de vehículos que pueden operar desde el depósito i .

2.3. Parámetros de los clientes

Para cada cliente $j \in J$:

- (x_j^c, y_j^c) : coordenadas del cliente j .
- q_j : demanda del cliente j (con $q_j > 0$).

2.4. Parámetros de vehículos y ruteo

- Q : capacidad de los vehículos (idéntica para todos los vehículos).
- g : costo fijo por ruta despachada.

2.5. Distancias

d_{ab} : distancia de viaje entre cualesquiera dos nodos $a, b \in I \cup J$.

Para las **instancias euclidianas** (escalas pequeña y mediana), las distancias por pares se calculan como:

$$d_{ab} = \text{round}\left(\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}, 1\right)$$

El redondeo a un decimal se aplica *por pares, antes de la suma*. El costo de una ruta es la suma de estos costos de arista redondeados; no se aplica ningún redondeo adicional a nivel de ruta o total. Esta regla es normativa y la hace cumplir el verificador oficial.

Para las **instancias explícitas** (escala grande), la matriz de distancias completa se proporciona directamente en el archivo de la instancia. Estas distancias se derivan del ruteo sobre la red vial real del Área Metropolitana de Monterrey y pueden ser asimétricas (es decir, $d_{ab} \neq d_{ba}$ en general). El verificador no supone la desigualdad triangular sobre las distancias explícitas.

3. Decisiones y restricciones

3.1. Decisiones

1. **Apertura de depósitos:** qué depósitos $i \in I$ abrir.
2. **Asignación de clientes:** cada cliente $j \in J$ debe asignarse a exactamente un depósito abierto.

3. **Diseño de rutas:** para cada depósito abierto, diseñar un conjunto de rutas de vehículos que atiendan a todos los clientes asignados a ese depósito. Cada ruta inicia y termina en su depósito asignado.

3.2. Restricciones

1. **Atención a los clientes:** cada cliente debe ser visitado exactamente una vez por exactamente una ruta.
2. **Asignación a depósitos:** cada ruta opera desde exactamente un depósito abierto, y todo cliente en esa ruta está asignado a ese depósito.
3. **Capacidad de los vehículos:** la demanda total de los clientes en cualquier ruta no debe exceder la capacidad del vehículo Q :

$$\sum_{j \in R_k} q_j \leq Q \quad \forall \text{ ruta } k$$

4. **Capacidad de los depósitos:** la demanda total de todos los clientes asignados al depósito i no debe exceder su capacidad W_i :

$$\sum_{j \in S_i} q_j \leq W_i \quad \forall i \in I \text{ con } y_i = 1$$

donde S_i es el conjunto de clientes asignados al depósito i y $y_i = 1$ indica que el depósito i está abierto.

5. **Límite de vehículos:** el número de rutas que operan desde el depósito i no debe exceder V_i :

$$|\{k : \text{la ruta } k \text{ opera desde el depósito } i\}| \leq V_i \quad \forall i \in I \text{ con } y_i = 1$$

3.3. Función objetivo

Minimizar el costo total:

$$\min \sum_{i \in I} f_i y_i + g \cdot |\mathcal{K}| + \sum_{k \in \mathcal{K}} \text{dist}(R_k)$$

donde:

- El primer término es el costo total de apertura de depósitos.
- El segundo término es el costo fijo total por despachar rutas (g por ruta, con \mathcal{K} el conjunto de todas las rutas).
- El tercer término es la distancia total de viaje, con $\text{dist}(R_k)$ la distancia de la ruta k (depósito \rightarrow primer cliente $\rightarrow \dots \rightarrow$ último cliente \rightarrow depósito).

4. Formatos de archivo

4.1. Formato de instancia

Los archivos de instancia utilizan formato de texto plano, delimitado por espacios en blanco. Las líneas que comienzan con # son comentarios y los analizadores las ignoran; se utilizan para transportar metadatos de procedencia (semilla del generador, versión, fecha del extracto de OSM).

```
# seed=<int>
# generator_version=<semver>
# osm_extract=<YYYY-MM-DD>           (solo para instancias FULL_MATRIX)
NAME : <instance_name>
CUSTOMERS : <n>
DEPOTS : <m>
VEHICLE_CAPACITY : <Q>
ROUTE_FIXED_COST : <g>
DISTANCE_FORMAT : COORDS | FULL_MATRIX
DEPOT_SECTION
<depot_id> <x> <y> <opening_cost> <capacity> <max_vehicles>
...
CUSTOMER_SECTION
<customer_id> <x> <y> <demand>
...
DISTANCE_SECTION
<matriz de distancias completa (m+n) x (m+n), fila por fila>
EOF
```

Notas:

- La DISTANCE_SECTION solo está presente cuando DISTANCE_FORMAT: FULL_MATRIX.
- Cuando DISTANCE_FORMAT: COORDS, las distancias son euclidianas (redondeadas por pares a un decimal antes de la suma).
- Los índices de los nodos en la matriz de distancias siguen el orden: depósitos $1, \dots, m$, luego clientes $m + 1, \dots, m + n$.
- Todos los valores son no negativos. Las demandas son enteros estrictamente positivos. Las distancias y los costos son reales no negativos.

4.2. Formato de solución

```
# instance=<instance_name>
COST : <total_cost>
DEPOTS_OPENED : <number_of_open_depots>
ROUTES : <total_number_of_routes>
DEPOT <depot_id>
  ROUTE : <customer_id_1> <customer_id_2> ... <customer_id_last>
  ROUTE : <customer_id_1> <customer_id_2> ...
DEPOT <depot_id>
  ROUTE : <customer_id_1> ...
...
EOF
```

Notas:

- Cada bloque DEPOT enumera las rutas que operan desde ese depósito.
- Cada línea ROUTE enumera los identificadores de clientes en orden de visita (el depósito está implícito al inicio y al final).
- El valor COST debe coincidir con el objetivo calculado a partir de la solución dentro de una tolerancia de 10^{-4} .
- Cada cliente debe aparecer exactamente una vez en el conjunto de todas las rutas.
- Las rutas vacías se rechazan. Los bloques de depósito sin rutas deben omitirse.

4.3. Verificación de soluciones

Una solución enviada es **factible** si y solo si se cumplen todas las siguientes condiciones:

1. Cada cliente aparece en exactamente una ruta.
2. Cada ruta está asignada a un depósito abierto.
3. La demanda total en cada ruta no excede Q .
4. La demanda total asignada a cada depósito abierto no excede W_i .
5. El número de rutas en cada depósito abierto no excede V_i .
6. El costo reportado coincide con el costo recalculado dentro de una tolerancia de 10^{-4} .

Una solución infactible se rechaza y no actualiza la tabla de clasificación.

5. Diseño de instancias

Las instancias se generarán mediante un generador de código abierto en Python y cubrirán las siguientes dimensiones:

Dimensión	Rango
Clientes (n)	200 – 3,000
Depósitos (m)	10 – 50
Número de instancias	30
Tipo de distancia	Euclidiana (pequeña/mediana); red vial de Monterrey, asimétrica (grande)

5.1. Variación estructural

Las instancias variarán a lo largo de los siguientes ejes para evitar la sobre-especialización:

- **Distribución de clientes:** aleatoria uniforme, agrupada, mixta (agrupamientos + uniforme).
- **Posicionamiento de depósitos:** en cuadrícula, aleatorio, co-ubicado con los agrupamientos de clientes, periférico.
- **Distribución de demanda:** uniforme, bimodal (clientes ligeros y pesados), proporcional al tamaño del agrupamiento.

- **Holgura de capacidad:** holgada (amplia capacidad de depósitos y vehículos), moderada, ajustada (la capacidad es una restricción activa).
- **Holgura del límite de vehículos:** holgada (los límites rara vez están activos) a ajustada (obliga a abrir depósitos).
- **Razón de costo de ruta:** g bajo (se prefieren muchas rutas cortas) a g alto (se prefieren menos rutas largas).

Para las instancias de gran escala, las ubicaciones de clientes y depósitos se muestrean a partir de polígonos reales del Área Metropolitana de Monterrey (zonas residenciales y comerciales para los clientes; zonas industriales para los depósitos), y las distancias se calculan mediante ruteo OSRM autoalojado sobre un extracto de Nuevo León de Geofabrik. Esto ancla el desafío en una geografía industrial genuinamente mexicana y refuerza la resonancia con el planteamiento de OXXO / Bimbo / FEMSA.

La distribución exacta de las instancias a lo largo de estos ejes se finalizará antes del lanzamiento de la competencia y se publicará en la configuración de la matriz de diseño del generador.

6. Formato de la competencia

6.1. Categorías

El desafío cuenta con dos categorías:

- a) **Categoría Licenciatura:** equipos de hasta 3 estudiantes de licenciatura vigentes más 1 guía (definido como cualquier persona que no sea estudiante de licenciatura vigente). Esta categoría se alinea con la misión educativa de la SMIO y fomenta la mentoría.
- b) **Categoría Abierta:** equipos de hasta 3 participantes, sin restricciones de perfil. Abierta a estudiantes, investigadores y profesionales de la industria.

Al menos la mitad de los integrantes del equipo debe contar con una membresía vigente de SMIO/A-LIO.

6.2. Puntuación y tabla de clasificación

6.2.1. Solución base de Hexaly y BKS de los participantes

Antes de que abra la competencia, Hexaly Optimizer se ejecuta sobre todas las instancias para producir una **solución base** para cada instancia. Estas soluciones base se publican a modo de referencia pero **no participan en la tabla de clasificación**. La tabla de clasificación registra únicamente las soluciones enviadas por los equipos participantes.

Esto significa que la mejor solución conocida (BKS, por sus siglas en inglés) en la tabla de clasificación siempre la ostenta un equipo participante, y el sistema de puntuación por tiempo de liderazgo (descrito más adelante) siempre produce una clasificación significativa, independientemente de si algún equipo mejora la solución base de Hexaly.

Al final de la competencia, los resultados se presentarán junto con la solución base de Hexaly, ofreciendo una comparación clara: para cada instancia se reportará si los participantes igualaron,

mejoraron o se quedaron por debajo de la solución de Hexaly. Esto preserva la narrativa atractiva del desafío, en la que la comunidad se mide frente a un potente solucionador industrial, garantizando al mismo tiempo que la competencia siempre tenga un ganador bien definido.

6.2.2. Puntuación por tiempo de liderazgo

Cada instancia tiene una BKS de los participantes que evoluciona durante la competencia, comenzando con el primer envío factible recibido.

Para cada instancia ℓ y cada equipo t , sea $\Delta_{\ell,t}$ el tiempo acumulado, medido en días, durante el cual el equipo t ostenta la BKS de la instancia ℓ . La medición es *deslizante*, con precisión a nivel de segundo: el cronómetro de un equipo inicia en la marca de tiempo exacta de recepción en el servidor del envío que establece su BKS, y se detiene en el instante exacto en que otro equipo la supera o en que cierra la competencia. Treinta y seis horas de liderazgo, por ejemplo, cuentan como 1.5 días. En caso de empate en el valor objetivo, el equipo en posesión conserva la BKS.

Sea $b_{\ell,t} = 1$ si el equipo t ostenta la BKS final de los participantes de la instancia ℓ al cierre de la competencia, y 0 en caso contrario.

La puntuación del equipo t es:

$$S_t = \sum_{\ell} \Delta_{\ell,t} + B \sum_{\ell} b_{\ell,t}$$

donde B es una bonificación (en días equivalentes) por ostentar la BKS final. El equipo con la puntuación total más alta gana en cada categoría. El valor de B se fija en 3 días.

6.2.3. Tablas de clasificación

Se mantendrán cuatro vistas públicas:

1. **Tabla de clasificación por instancia:** para cada instancia, el costo de la BKS actual de los participantes, el equipo que la ostenta y la marca de tiempo en que se estableció.
2. **Tabla de clasificación global:** la puntuación agregada S_t de cada equipo, actualizada en tiempo real.
3. **Comparación con Hexaly:** para cada instancia, el costo de la solución base de Hexaly junto con la BKS actual de los participantes, con la brecha porcentual entre ambas.
4. **Detalle por instancia:** para cada instancia, sus estadísticas básicas (escala, número de clientes y depósitos) y la evolución cronológica completa de su BKS de los participantes durante la ventana: qué equipo la ostentó, a qué costo y entre qué instantes. Los archivos de solución en sí mismos y los metadatos por envío (tiempo de cómputo, entorno) permanecen reservados hasta el cierre de la competencia.

El costo de la BKS actual y el equipo que la ostenta son públicos en tiempo real. Los archivos de solución reales se mantienen reservados hasta el final de la competencia, lo que impide que los equipos mejoren localmente los envíos de los demás.

Cuando un envío establece una nueva BKS, se solicita al equipo remitente (tras la verificación) que proporcione metadatos adicionales sobre ese envío específico: el tiempo de cómputo de reloj de pared (wall-clock) en segundos y una breve descripción del entorno computacional (versiones

de software, CPU, RAM). Estos metadatos se registran junto con la solución en la biblioteca de referencia.

6.3. Descripción del método

Cada equipo envía un PDF conciso que describe su enfoque al momento del registro del equipo. El documento abarca: la reformulación del problema (si la hay), una descripción general del método, las herramientas y solucionadores, el hardware y el presupuesto computacional. La descripción es revisada por el Consejo Directivo de la SMIO como parte del proceso de aprobación del registro. Al final de la competencia, las descripciones de método de los equipos participantes se publican en la biblioteca de referencia, junto con las soluciones con las que cada uno estableció una BKS, transformando el artefacto de un conjunto de datos en una colección citable de enfoques.