

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INDUSTRIAS

PROPUESTA DE MEJORA PARA EL SISTEMA DE TRANSPORTE PÚBLICO PARA
LA CONEXIÓN DE COLINA-SANTIAGO

AUTOR
JOEL GABRIEL FIGUEROA GALARZA

PROFESOR GUÍA
ESTEBAN SALGADO VALENZUELA

PROFESOR CORREFERENTE
ISMAEL KAUAK VERDUGO

SANTIAGO, 20 DE AGOSTO, 2026



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☐ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: Propuesta de mejora para el sistema de transporte público para la conexión de Celina - Santiago

Nombre del candidato(a): Joel Gabriel Figueira Galarza

Carrera / Grado: Ingeniería Civil Industrial

Campus: Vitacura Departamento: Industrias

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Esteban Salgado Valenzuela, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (marcar una opción):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 20/08/2026

Firma: 

Estudiante o Candidato(a):

Fecha: 18/08/26

Firma: Joel Figueira

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Agradecimientos

Quiero agradecer, en primer lugar, a mis padres, por todos estos años de apoyo incondicional. Gracias por confiar en mí, por nunca presionarme y por acompañarme en cada etapa de este proceso. Espero haber podido retribuir ese esfuerzo logrando terminar y llegar hasta aquí sin que tuvieran que preocuparse demasiado.

Agradezco también a mi pareja, Antonia, futura arquitecta, por estar conmigo casi desde el comienzo de este camino. Gracias por apoyarme y entenderme incluso en la distancia y a pesar del poco tiempo que a veces teníamos disponible. Juntos pusimos siempre la responsabilidad por delante y logramos llegar hasta el final. Espero que tengamos un futuro próspero por delante.

Por último, quiero agradecer a mi gran grupo de amigos, "Los Prestigios". Estoy convencido de que somos el mejor grupo de la generación, uno que nunca se separó y que pronto se titulará en conjunto como futuros ingenieros. Gracias por el apoyo incondicional que siempre encontré en ustedes, tanto dentro de la universidad como en los ámbitos más personales de la vida.

Índice

1. Problema de investigación	7
2. Objetivos	10
2.1. Objetivo General	10
2.2. Objetivos Específicos	10
3. Marco Teórico	12
3.1. Logística, transporte y movilidad interurbana	12
3.1.1. Logística y transporte como soporte territorial	12
3.1.2. Transporte y movilidad interurbana	13
3.2. Modelos de optimización matemática	13
3.2.1. Definición y propósito de los modelos de optimización	13
3.2.2. Programación lineal y programación entera mixta	15
3.2.3. Componentes de un modelo de optimización	16
3.3. Modelos de localización de puntos de acceso al transporte	18
3.3.1. Localización de instalaciones en sistemas logísticos y de transporte	18
3.3.2. Modelos de cobertura	19
3.3.3. Modelos de p-mediana y minimización de distancia	20
3.3.4. Aplicación a puntos de acceso al transporte	21
3.4. Modelos de redes y ruteo en transporte	22
3.4.1. Representación de redes mediante nodos y arcos	22
3.4.2. Divergencia y conservación de flujo en redes	23
3.4.3. Problemas de ruta mínima	24
3.4.4. Problemas de ruteo de vehículos	25
3.4.5. Rutas troncales y rutas de alimentación	26
3.4.6. Restricciones de capacidad, tiempo y cobertura	27
3.4.7. OpenRouteService como herramienta de cálculo de rutas sobre la red vial	28

3.5.	Caracterización del sistema de transporte Colina-Santiago	29
3.5.1.	Contexto territorial y crecimiento urbano de Colina	29
3.5.2.	Dependencia funcional entre Colina y Santiago	30
3.5.3.	Corredores principales de conexión	31
3.5.4.	Transporte público, transporte privado y patrones de movilidad . .	31
4.	Metodología	32
4.1.	Formulación del modelo de localización	33
4.1.1.	Análisis del problema y estrategia de resolución	39
4.1.2.	Construcción de la base de datos georreferenciada	42
4.1.3.	Procesamiento espacial y construcción de vecindades	45
4.1.4.	Generación de parámetros del modelo	46
4.1.5.	Implementación y resolución del modelo en Pyomo	51
4.1.6.	Análisis de resultados	52
4.2.	Formulación del modelo de ruteo	53
4.2.1.	Preparación de los insumos del modelo de ruteo	58
4.2.2.	Construcción de la matriz de tiempos y arcos factibles	59
4.2.3.	Tratamiento de pasajes sin salida y coherencia espacial	60
4.2.4.	Procedimiento de resolución computacional	61
4.2.5.	Evaluación de las rutas generadas	62
5.	Resultados y discusión	62
5.1.	Resultados de la matriz de distancias	62
5.2.	Resultados del modelo de localización	64
5.3.	Comparación espacial de cobertura	65
5.4.	Resultados del modelo de ruteo	68
5.5.	Análisis operacional de las rutas propuestas	73
5.6.	Segunda iteración: corrección de paraderos en pasajes sin salida	75
5.6.1.	Efecto de la corrección de pasajes sobre la cobertura de demanda	80

5.7. Discusión integrada de resultados	81
5.8. Comparación cuantitativa entre la situación actual y la propuesta	84
6. Conclusión y recomendaciones	86
6.1. Conclusión	86
6.2. Recomendaciones	89
7. Referencias	91
8. Anexos	94
8.1. Código para la construcción de la matriz de distancias y tiempos	94
8.2. Código para resolver el modelo de localización	98
8.3. Código para resolver el modelo de ruteo	103

Índice de tablas

1.	Parámetros base utilizados en el modelo de localización	51
2.	Resumen de insumos espaciales del modelo de ruteo	58
3.	Parámetros principales utilizados en el modelo de ruteo	59
4.	Criterios utilizados para el diagnóstico de pasajes sin salida	60
5.	Parámetros computacionales de resolución	61
6.	Resumen de la matriz de distancias	63
7.	Resumen de resultados por escenario	64
8.	Resumen del procesamiento previo al modelo de ruteo	68
9.	Resumen de rutas consideradas en la propuesta	69
10.	Resumen del procesamiento previo a la segunda iteración	75
11.	Resumen de rutas de la segunda iteración	75
12.	Comparación entre la primera y la segunda iteración	75
13.	Efecto de la exclusión de paraderos sobre la cobertura de demanda	81
14.	Comparación cuantitativa entre la situación actual y la propuesta	85

Índice de imágenes

1.	Mapa base del sector Colina centro y Esmeralda.	43
2.	Puntos candidatos de recolección de pasajeros en el sector Colina centro y Esmeralda.	44
3.	Nodos de demanda representados en el sector Colina centro y Esmeralda.	44
4.	Distribución espacial de los paraderos actuales	66
5.	Distribución espacial de los puntos seleccionados en el escenario propuesto ($p = 60$)	67
6.	Trazado generado para la Ruta 1	70
7.	Trazado generado para la Ruta 2	71
8.	Trazado correspondiente a la Ruta 3, incorporada como ruta fija actual . .	72
9.	Esquema conjunto de las rutas propuestas y su conexión con Santiago mediante la Autopista Los Libertadores hasta la estación Los Libertadores	73
10.	Trazado generado para la Ruta 1, segunda iteración	77
11.	Trazado generado para la Ruta 2, segunda iteración	78
12.	Trazado correspondiente a la Ruta 3, segunda iteración	79
13.	Esquema conjunto de las rutas propuestas en la segunda iteración	80

1. Problema de investigación

La movilidad interurbana entre Colina y Santiago constituye un desafío relevante para la planificación del transporte en la Región Metropolitana. La comuna ha experimentado un crecimiento urbano y demográfico sostenido, junto con una integración funcional cada vez mayor con la capital, debido a la concentración de empleos, servicios, educación y actividades comerciales en Santiago. Esta situación ha incrementado los desplazamientos cotidianos hacia el área metropolitana y ha generado mayor presión sobre los principales corredores de conexión, como Autopista Los Libertadores, Autopista Nororiente y caleta General San Martín.

Según el Plan Maestro de Transporte Santiago 2025 del Ministerio de Transportes y Telecomunicaciones, la expansión metropolitana y el aumento de la motorización han intensificado la demanda sobre los accesos a la ciudad, afectando especialmente a comunas periféricas que dependen de corredores interurbanos para conectarse con el centro metropolitano. En el caso de Colina, esta presión se ve reforzada por su crecimiento poblacional. De acuerdo con el Reporte Comunal de la Biblioteca del Congreso Nacional de Chile, la comuna pasó de 146.207 habitantes en 2017 a una proyección de 197.370 habitantes en 2024, lo que representa una variación de 35,0 %. Este aumento ha elevado la demanda potencial de viajes y ha generado mayores exigencias sobre la infraestructura vial y la oferta de transporte público.

El crecimiento territorial de Colina no se ha desarrollado de manera homogénea. El Plan de Desarrollo Comunal 2020-2030 identifica una configuración urbana heterogénea, con sectores de mayor concentración urbana, como Colina centro, y áreas de expansión más dispersas, como Chicureo, Liray y Santa Elena. Esta distribución dificulta la provisión de transporte público bajo esquemas tradicionales, ya que la demanda no se concentra en un único eje ni en un solo punto de origen. Por el contrario, los usuarios se distribuyen en distintos sectores que requieren soluciones diferenciadas de acceso, recolección y conexión con los corredores principales.

En este contexto, el problema de movilidad de Colina no se explica únicamente

por la distancia hacia Santiago, sino por la baja articulación entre la oferta de transporte público y la distribución espacial de la demanda. La comuna cuenta con corredores viales estructurantes hacia la capital, pero el desafío principal se encuentra en la etapa previa al viaje interurbano: cómo acercar a los usuarios desde sus sectores de origen, cómo organizar la demanda en puntos adecuados de recolección y cómo conectar dichos puntos con servicios de mayor capacidad hacia Santiago.

El transporte público que opera actualmente en el eje Colina-Santiago presenta limitaciones en cobertura, frecuencia, regularidad, costo e integración. Según antecedentes del PLADECO, el transporte público mayor de la comuna se compone principalmente de servicios de buses rurales con estándares variables, baja cobertura en algunos sectores, menor frecuencia fuera de los horarios punta y escasa infraestructura formal de paraderos. Además, la falta de integración tarifaria con el sistema metropolitano reduce la competitividad del transporte público frente al automóvil particular, especialmente para quienes deben realizar viajes diarios.

Esta baja competitividad ha favorecido una alta dependencia del vehículo privado. Un sondeo realizado por Chicureo Hoy indica que un 63,9 % de los vecinos se moviliza en automóvil particular, mientras que solo un 25,3 % utiliza buses públicos. Asimismo, las rutas más utilizadas corresponden a Autopista Los Libertadores, con un 41 %, y Radial Nororiente, con un 40 %, lo que evidencia una fuerte concentración de los desplazamientos en los principales accesos viales hacia Santiago. Esta situación intensifica la congestión, aumenta la variabilidad de los tiempos de viaje y afecta tanto a usuarios del transporte privado como del transporte público.

Desde una perspectiva operacional, el sistema presenta una tensión entre conectividad macro y accesibilidad local. Si bien Colina dispone de autopistas y vías principales que permiten la conexión con Santiago, muchos sectores interiores dependen de vías secundarias, accesos limitados o recorridos poco directos para incorporarse a dichos corredores. En consecuencia, el tiempo total de viaje no depende solo del tramo interurbano, sino también del tiempo de acceso al servicio, la espera, los transbordos y la regularidad de la

operación. Esto resulta especialmente relevante considerando que el PLADECO informa un viaje promedio de 43 minutos en horario punta tarde para la comuna.

A partir de estos antecedentes, la problemática central corresponde a la necesidad de mejorar la organización operacional del transporte público interurbano entre Colina y Santiago. En particular, se requiere analizar cómo recolectar la demanda desde las principales zonas de la comuna, dónde ubicar puntos estratégicos de consolidación y cómo estructurar rutas alimentadoras que atiendan la demanda bajo condiciones adecuadas de capacidad y nivel de servicio, sentando las bases para un futuro dimensionamiento de frecuencia y flota.

Por lo tanto, el proyecto se orienta a desarrollar una propuesta de mejora basada en herramientas de investigación de operaciones y análisis territorial. El sistema puede representarse como una red compuesta por zonas de demanda, puntos candidatos de recolección, corredores de conexión, vehículos y restricciones operacionales. Bajo esta lógica, es posible evaluar alternativas de localización, cobertura, ruteo y capacidad que permitan mejorar la eficiencia, accesibilidad y confiabilidad del transporte público en el eje Colina-Santiago.

En función de lo anterior, se formulan las siguientes preguntas de investigación:

¿Cómo puede optimizarse la recolección de pasajeros en las principales zonas de Colina para mejorar la conexión con Santiago?

¿Qué puntos de consolidación permitirían aumentar la cobertura del transporte público y reducir los tiempos de acceso de los usuarios?

¿Qué estructura de rutas alimentadoras resulta más eficiente para conectar Colina centro con los corredores troncales hacia Santiago?

¿De qué manera la propuesta puede mejorar la accesibilidad al transporte público y generar condiciones que favorezcan su uso frente al automóvil particular en la conexión Colina-Santiago?"

2. Objetivos

2.1. Objetivo General

Diseñar una propuesta de mejora operacional para el sistema de transporte público interurbano entre Colina y Santiago, centrada en la localización de puntos de recolección de pasajeros mediante un modelo de optimización de cobertura, y complementada con una propuesta preliminar de rutas alimentadoras que conecten dichos puntos con los corredores troncales hacia Santiago, con el fin de mejorar la cobertura territorial y la accesibilidad peatonal al sistema.

2.2. Objetivos Específicos

Caracterizar el sistema actual de transporte público interurbano entre Colina y Santiago, considerando sus corredores de conexión, puntos de detención existentes, cobertura territorial y condiciones de accesibilidad, con el fin de identificar las limitaciones operacionales que justifican la propuesta.

Analizar la distribución espacial de la demanda potencial en la comuna de Colina, identificando los nodos de demanda asociados a las zonas residenciales y su relación con los corredores hacia Santiago, a partir de una base de datos georreferenciada.

Construir las matrices de distancia y tiempo de caminata entre los nodos de demanda y los puntos candidatos de recolección sobre la red vial, empleando OpenRouteService, para disponer de los parámetros que alimentan el modelo de localización.

Formular e implementar el modelo de Cobertura Máxima (Church y ReVelle, 1974) para seleccionar los puntos de recolección de pasajeros, incorporando la asignación de demanda, la capacidad de cada punto, el número de puntos a habilitar y un estándar mínimo de cobertura del 90 %.

Evaluar la solución de localización mediante indicadores de cobertura, demanda cubierta, utilización de los puntos habilitados y distribución espacial, comparando la propuesta con la situación actual de paraderos.

Proponer, como etapa complementaria, un conjunto preliminar de rutas alimentadoras que conecten los puntos de recolección seleccionados con un acceso a los corredores troncales, mediante un modelo de ruteo con restricción de tiempo máximo, entregando trazados base para orientar una eventual definición operacional posterior.

3. Marco Teórico

El presente marco teórico tiene como propósito establecer los fundamentos conceptuales, metodológicos y matemáticos que sustentan el análisis del sistema de transporte público interurbano entre Colina y Santiago.

El sistema Colina-Santiago puede ser comprendido como una red de transporte compuesta por zonas de generación de viajes, puntos de acceso, corredores viales, vehículos, usuarios y restricciones operacionales. Esta red presenta una estructura territorial heterogénea, ya que combina zonas de mayor concentración urbana, como Colina centro, con sectores de menor densidad o mayor dispersión, como Liray, Chicureo y Santa Elena. Esta diferencia territorial genera desafíos relevantes para el diseño del transporte público, debido a que una ruta única o uniforme no necesariamente responde de manera eficiente a la localización real de la demanda.

3.1. Logística, transporte y movilidad interurbana

3.1.1. Logística y transporte como soporte territorial

La logística puede entenderse como el proceso encargado de planificar, implementar y controlar el flujo de bienes, servicios e información desde un punto de origen hasta un punto de destino, buscando cumplir con los requerimientos del usuario de forma eficiente. Dentro de este sistema, el transporte cumple un rol central, debido a que permite materializar físicamente dichos flujos y conectar actividades distribuidas en el territorio. Ballou (2004) señala que las decisiones de transporte y localización inciden directamente en los costos, tiempos y niveles de servicio de un sistema logístico, por lo que su análisis resulta fundamental para comprender el funcionamiento de una red.

Aunque la logística suele asociarse al movimiento de mercancías, sus principios también permiten interpretar sistemas de movilidad de personas. En ambos casos existe una demanda que debe ser atendida, una red de conexión, recursos limitados y restricciones

operacionales. Bajo esta perspectiva, los desplazamientos de pasajeros pueden analizarse como flujos que se distribuyen sobre una red territorial, donde la eficiencia del sistema depende de la relación entre infraestructura, capacidad, recorridos, frecuencia y localización de los puntos de acceso.

3.1.2. Transporte y movilidad interurbana

El transporte corresponde al componente de la logística que permite conectar físicamente distintos puntos de una red. En el caso de la movilidad de personas, este cumple una función social y territorial, ya que permite acceder a empleo, educación, comercio, salud y otros servicios. Por ello, un sistema de transporte no debe evaluarse únicamente por la existencia de caminos o corredores viales, sino también por la calidad del servicio, la cobertura, los tiempos de viaje y la facilidad con que los usuarios pueden acceder a la red.

La movilidad interurbana se refiere a los desplazamientos realizados entre zonas urbanas, semiurbanas o rurales que mantienen una relación funcional entre sí. En comunas periféricas, estos viajes suelen estar asociados a la dependencia respecto de centros metropolitanos mayores, donde se concentran actividades y servicios especializados. En este sentido, SOCHITRAN (2013) plantea que las políticas de transporte deben abordarse considerando la integración territorial y la calidad del sistema, especialmente en contextos donde existen diferencias de acceso entre distintos sectores de la ciudad.

3.2. Modelos de optimización matemática

3.2.1. Definición y propósito de los modelos de optimización

La investigación de operaciones entrega herramientas para representar problemas de transporte mediante modelos matemáticos que apoyan la toma de decisiones. En este ámbito, los modelos permiten analizar la localización de puntos de acceso, la asignación de demanda, el diseño de recorridos, la utilización de capacidad y la evaluación de alternativas

operacionales. Su utilidad radica en transformar un problema territorial en una estructura formal, donde las decisiones se expresan mediante variables, las condiciones del sistema mediante restricciones y el objetivo del análisis mediante una función a optimizar.

Los modelos clásicos de transporte desarrollados por Hitchcock (1941) y Koopmans (1949) constituyen una base relevante para este tipo de análisis, ya que permiten representar la distribución de flujos entre puntos de origen y destino bajo restricciones de oferta y demanda. Aunque estos modelos fueron planteados inicialmente para problemas de distribución de productos, su lógica puede extenderse al transporte de pasajeros, debido a que ambos casos consideran una demanda que debe ser atendida mediante una red de conexión y recursos limitados.

De manera general, un problema de transporte puede representarse mediante un conjunto de orígenes I , un conjunto de destinos J , una oferta o disponibilidad s_i en cada origen, una demanda d_j en cada destino, un costo c_{ij} asociado al traslado entre i y j , y una variable x_{ij} que representa el flujo enviado desde el origen i hacia el destino j . Una formulación básica corresponde a:

$$\text{mín} \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

sujeto a:

$$\sum_{j \in J} x_{ij} \leq s_i \quad \forall i \in I$$

$$\sum_{i \in I} x_{ij} \geq d_j \quad \forall j \in J$$

$$x_{ij} \geq 0 \quad \forall i \in I, \forall j \in J$$

En el contexto del eje Colina-Santiago, esta estructura permite interpretar los sectores de la comuna como puntos de origen de demanda y los puntos de acceso o

conexión como destinos intermedios dentro del sistema. De esta forma, el modelo permite analizar cómo asignar usuarios a puntos de recolección, considerando costos de acceso, disponibilidad de capacidad y distribución espacial de la demanda.

3.2.2. Programación lineal y programación entera mixta

La programación lineal corresponde a una técnica de optimización matemática en la que tanto la función objetivo como las restricciones se expresan mediante relaciones lineales. Este tipo de modelos permite representar problemas donde se busca asignar recursos limitados de manera eficiente, considerando condiciones de disponibilidad, demanda, capacidad o balance entre distintos elementos del sistema. En el ámbito del transporte, la programación lineal resulta útil para analizar flujos, asignaciones y distribución de recursos dentro de una red.

De manera general, un modelo de programación lineal puede expresarse como:

$$\text{mín } Z = c^T x$$

sujeto a:

$$Ax \leq b$$

$$x \geq 0$$

donde x representa el vector de variables de decisión, c corresponde al vector de costos o coeficientes de la función objetivo, A representa la matriz de coeficientes técnicos y b contiene los límites, recursos disponibles o capacidades del sistema. Esta estructura permite evaluar soluciones factibles y seleccionar aquella que obtiene el mejor valor de la función objetivo.

Sin embargo, existen problemas en los que algunas decisiones no pueden represen-

tarse únicamente mediante variables continuas, ya que requieren seleccionar alternativas, activar componentes del sistema o definir cantidades enteras. Para estos casos se utiliza la programación entera mixta, la cual combina variables continuas con variables enteras o binarias. Una variable binaria puede representarse de manera general como:

$$y_j = \begin{cases} 1, & \text{si la alternativa } j \text{ es seleccionada} \\ 0, & \text{en caso contrario} \end{cases}$$

Este tipo de formulación permite incorporar decisiones discretas dentro de un modelo de optimización, manteniendo al mismo tiempo variables continuas asociadas a flujos, cantidades, costos o tiempos. Por esta razón, la programación entera mixta resulta especialmente relevante en problemas de transporte, donde suelen coexistir decisiones de asignación, selección, capacidad y operación. Su uso permite representar sistemas en los que las decisiones estratégicas y operacionales se encuentran relacionadas dentro de una misma estructura matemática.

3.2.3. Componentes de un modelo de optimización

Un modelo de optimización matemática se compone, en términos generales, de una función objetivo, variables de decisión, parámetros y restricciones. La función objetivo representa el criterio que se busca optimizar, ya sea minimizar costos, reducir tiempos, disminuir distancias o maximizar cobertura. Las variables de decisión corresponden a los elementos que el modelo debe determinar, mientras que los parámetros representan información conocida del sistema, como demanda, distancias, capacidades o costos. Finalmente, las restricciones establecen las condiciones que toda solución debe cumplir para ser considerada factible.

De manera general, un problema de optimización puede representarse como:

$$\text{mín } Z = f(x)$$

sujeto a:

$$g_m(x) \leq b_m \quad \forall m \in M$$

$$x \in X$$

donde Z corresponde al valor de la función objetivo, x representa el conjunto de variables de decisión, $f(x)$ es la función que se desea minimizar o maximizar, $g_m(x)$ representa las restricciones del problema, b_m corresponde a los límites o recursos disponibles, y X define el dominio de las variables.

En problemas de transporte, estos componentes permiten representar decisiones asociadas a la asignación de demanda, utilización de capacidad, selección de alternativas y organización de flujos dentro de una red. Su importancia radica en que entregan una estructura formal para analizar sistemas donde existen recursos limitados, múltiples restricciones y objetivos que pueden entrar en conflicto entre sí.

La optimización matemática permite estudiar estos sistemas bajo criterios definidos previamente, estableciendo una relación entre las decisiones disponibles y las condiciones que limitan su implementación. De esta manera, el problema deja de abordarse solo como una descripción operacional y pasa a representarse como un conjunto de relaciones matemáticas que permiten evaluar la factibilidad y eficiencia de distintas soluciones.

En el ámbito del transporte, esta aproximación resulta relevante porque los cambios en un componente del sistema pueden afectar el desempeño de los demás. La capacidad disponible, la distribución de la demanda, los tiempos de viaje y las condiciones de operación se relacionan entre sí, por lo que requieren ser analizados de manera conjunta dentro de una misma estructura de decisión.

3.3. Modelos de localización de puntos de acceso al transporte

3.3.1. Localización de instalaciones en sistemas logísticos y de transporte

La localización de instalaciones o puntos de acceso corresponde a una decisión estratégica dentro de los sistemas logísticos y de transporte, debido a que define en qué puntos del territorio se ubicarán servicios, nodos de atención o puntos de conexión para atender una demanda distribuida espacialmente. En términos generales, este tipo de decisiones busca mejorar el nivel de servicio del sistema mediante la reducción de distancias, tiempos de acceso, costos de operación o zonas sin cobertura. Desde la perspectiva logística, la localización no se analiza de forma aislada, ya que se relaciona directamente con las decisiones de transporte, asignación de demanda y utilización de recursos disponibles (Ballou, 2004).

En el transporte de pasajeros, los puntos de acceso pueden entenderse como nodos que permiten vincular a los usuarios con una red principal de movilidad. Estos puntos pueden corresponder a paraderos, estaciones, terminales menores, zonas de transferencia o puntos de recolección. Su ubicación es relevante porque determina la distancia que deben recorrer los usuarios para ingresar al sistema, la cantidad de demanda que puede ser captada y la eficiencia de los recorridos posteriores. Por lo tanto, una localización inadecuada puede aumentar los tiempos de acceso, reducir la cobertura efectiva y concentrar innecesariamente la demanda en ciertos sectores.

De manera general, los problemas de localización consideran una demanda distribuida en el territorio y un conjunto de ubicaciones posibles para instalar puntos de servicio. La decisión consiste en seleccionar aquellas ubicaciones que permitan atender dicha demanda bajo ciertos criterios, como cobertura, distancia, tiempo de acceso, capacidad o cantidad máxima de puntos disponibles. Esta representación permite analizar el territorio como un sistema de demanda y oferta espacial, donde la ubicación de cada punto incide en el desempeño general de la red.

3.3.2. Modelos de cobertura

Los modelos de cobertura buscan determinar la ubicación de instalaciones de manera que la mayor cantidad posible de demanda quede atendida dentro de una distancia o tiempo máximo aceptable. Este enfoque es útil cuando no basta con ubicar puntos cercanos al centro del territorio, sino que se requiere asegurar que los usuarios se encuentren dentro de un rango razonable de acceso. Church y ReVelle (1974) desarrollan el problema de localización de máxima cobertura, el cual permite seleccionar un número limitado de instalaciones con el objetivo de maximizar la demanda cubierta.

Una formulación general del modelo de máxima cobertura puede expresarse como:

$$\text{máx} \sum_{i \in I} d_i y_i$$

sujeto a:

$$\sum_{j \in N_i} x_j \geq y_i \quad \forall i \in I$$

$$\sum_{j \in J} x_j = p$$

$$x_j \in \{0, 1\} \quad \forall j \in J$$

$$y_i \in \{0, 1\} \quad \forall i \in I$$

donde y_i toma el valor de 1 si la zona de demanda i queda cubierta, x_j toma el valor de 1 si el punto candidato j es seleccionado, p representa la cantidad máxima de puntos a instalar y N_i corresponde al conjunto de puntos candidatos que se encuentran dentro del radio o tiempo máximo aceptable para cubrir la zona i .

Aplicado al transporte de pasajeros, este modelo permite analizar qué puntos de

recolección logran cubrir una mayor proporción de usuarios dentro de una distancia de caminata o traslado inicial aceptable. Esto resulta especialmente relevante en comunas con demanda dispersa, donde seleccionar puntos de acceso sin un criterio de cobertura puede dejar sectores sin atención efectiva o forzar desplazamientos iniciales demasiado largos.

3.3.3. Modelos de p-mediana y minimización de distancia

Los modelos de mediana buscan ubicar instalaciones de manera que se minimice la distancia total ponderada entre los puntos de demanda y las instalaciones seleccionadas. A diferencia de los modelos de cobertura, que se enfocan en cumplir una distancia o tiempo máximo aceptable, los modelos de mediana priorizan reducir el esfuerzo total de desplazamiento de los usuarios. Hakimi (1964) plantea bases relevantes para este tipo de problemas en redes, mostrando que la localización óptima puede estudiarse considerando la estructura del grafo y la distribución de la demanda.

Una formulación general del problema de p mediana puede expresarse como:

$$\text{mín} \sum_{i \in I} \sum_{j \in J} d_i c_{ij} z_{ij}$$

sujeto a:

$$\sum_{j \in J} z_{ij} = 1 \quad \forall i \in I$$

$$z_{ij} \leq x_j \quad \forall i \in I, \forall j \in J$$

$$\sum_{j \in J} x_j = p$$

$$x_j \in \{0, 1\} \quad \forall j \in J$$

$$z_{ij} \in \{0, 1\} \quad \forall i \in I, \forall j \in J$$

donde z_{ij} toma el valor de 1 si la zona de demanda i es asignada al punto j , c_{ij} representa la distancia o tiempo de acceso entre ambos puntos, d_i corresponde a la demanda de la zona i , y p indica la cantidad de puntos de acceso que se desea seleccionar.

En el contexto del transporte público, este enfoque permite evaluar ubicaciones que reduzcan la distancia promedio o total que deben recorrer los usuarios para acceder al sistema. Por ello, resulta útil cuando el objetivo principal no es solamente cubrir sectores, sino disminuir el esfuerzo agregado de acceso y mejorar la eficiencia territorial de los puntos de abordaje.

3.3.4. Aplicación a puntos de acceso al transporte

Para el eje Colina-Santiago, los modelos de localización permiten justificar técnicamente la selección de puntos de recolección dentro de la comuna. Estos puntos deben ubicarse considerando la distribución espacial de la demanda, la estructura vial local, la cercanía a los corredores principales y la posibilidad de integrarse a recorridos internos de alimentación. De esta forma, la localización no se entiende solo como una decisión geográfica, sino como una decisión operacional que afecta la cobertura, los tiempos de acceso y la eficiencia del sistema completo.

En este caso, la aplicación de modelos de cobertura permitiría identificar qué puntos candidatos logran atender a una mayor proporción de usuarios dentro de un radio o tiempo máximo aceptable. Por otra parte, la aplicación de modelos de mediana permitiría seleccionar ubicaciones que reduzcan la distancia total ponderada entre los sectores de demanda y los puntos de recolección. Ambos enfoques pueden complementarse, ya que el primero asegura cobertura territorial y el segundo mejora la eficiencia del acceso.

Esta lógica resulta pertinente en sistemas donde la demanda se encuentra distribuida en distintos sectores del territorio y no converge de forma natural hacia un único nodo.

En estos casos, la selección de puntos de acceso debe evitar la concentración excesiva de usuarios en un solo lugar, ya que esto puede generar mayores tiempos de acceso, pérdida de cobertura efectiva y una operación menos eficiente del sistema. En cambio, una red de puntos distribuidos permite organizar mejor la demanda antes de conectarla con corredores principales o servicios de mayor jerarquía.

3.4. Modelos de redes y ruteo en transporte

3.4.1. Representación de redes mediante nodos y arcos

Una red de transporte puede representarse como un grafo compuesto por un conjunto de nodos y arcos. Los nodos corresponden a puntos relevantes del sistema, tales como zonas de origen, puntos de recolección, intersecciones, paraderos o destinos. Los arcos representan las conexiones entre dichos nodos, como calles, caminos, autopistas o tramos de recorrido. Esta representación permite estudiar el transporte como una estructura organizada, donde los desplazamientos se realizan sobre una red con distancias, tiempos de viaje y capacidades definidas.

De manera general, una red puede expresarse como:

$$G = (N, A)$$

donde N representa el conjunto de nodos y A el conjunto de arcos que conectan dichos nodos. Para cada arco $(i, j) \in A$, se puede asociar un costo c_{ij} , que puede representar distancia, tiempo de viaje o costo operacional. Además, en algunos casos se puede incorporar una capacidad u_{ij} , la cual limita el flujo máximo que puede circular por dicho arco.

En el caso de Colina, los nodos pueden asociarse a sectores generadores de demanda, puntos de acceso al transporte y conexiones hacia los corredores principales. Por su parte, los arcos corresponden a las vías internas de la comuna y a los corredores de conexión hacia Santiago. Esta formulación permite analizar el sistema como una red de

movilidad, donde la demanda debe desplazarse desde distintos sectores de origen hacia puntos de recolección o rutas principales bajo restricciones de distancia, tiempo y capacidad.

3.4.2. Divergencia y conservación de flujo en redes

En una red de transporte, la divergencia permite representar el balance de flujo en cada nodo. Este concepto mide la diferencia entre el flujo que sale desde un nodo y el flujo que entra hacia él. Si se define x_{ij} como el flujo que circula por el arco (i, j) , la divergencia en un nodo i puede expresarse como:

$$\sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = b_i \quad \forall i \in N$$

donde el primer término representa el flujo total que sale desde el nodo i , el segundo término representa el flujo total que entra al nodo i , y b_i corresponde al balance neto del nodo. Si $b_i > 0$, el nodo actúa como origen o generador de flujo. Si $b_i < 0$, el nodo actúa como destino o receptor de flujo. Si $b_i = 0$, el nodo funciona como punto intermedio o de transbordo, por lo que todo flujo que entra debe salir.

Este concepto es fundamental en los modelos de redes, ya que permite asegurar que los flujos se conserven dentro del sistema. En transporte de pasajeros, esto implica que la cantidad de usuarios que se desplaza por la red debe mantenerse coherente entre los distintos nodos, salvo en aquellos puntos donde se genera demanda, se recoge pasajeros o se produce una transferencia hacia otro servicio.

En el caso de Colina, los sectores que generan viajes pueden representarse como nodos con divergencia positiva, mientras que los puntos de recolección, corredores principales o destinos hacia Santiago pueden representarse como nodos receptores o de transferencia. De esta manera, la divergencia permite modelar la continuidad de los desplazamientos desde los sectores internos de la comuna hacia la red principal de conexión.

3.4.3. Problemas de ruta mínima

Los problemas de ruta mínima buscan determinar el camino de menor costo, distancia o tiempo entre dos nodos de una red. Este tipo de problema es relevante en transporte, ya que permite identificar conexiones eficientes entre sectores de origen y puntos de destino. Dijkstra (1959) propone uno de los algoritmos clásicos para resolver este problema en grafos con costos no negativos.

Una formulación básica del problema de ruta mínima entre un nodo de origen s y un nodo de destino t puede expresarse como:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij}$$

sujeto a:

$$\sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = \begin{cases} 1, & \text{si } i = s \\ -1, & \text{si } i = t \\ 0, & \text{en otro caso} \end{cases} \quad \forall i \in N$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A$$

En esta formulación, c_{ij} representa el costo de utilizar el arco (i, j) , mientras que x_{ij} indica si dicho arco forma parte de la ruta seleccionada. La restricción de balance asegura que el flujo salga desde el nodo de origen, llegue al nodo de destino y se conserve en los nodos intermedios.

En el contexto del eje Colina-Santiago, este enfoque puede utilizarse para evaluar conexiones internas dentro de la comuna o para analizar el acceso desde ciertos sectores hacia los corredores principales. Sin embargo, la ruta más corta no siempre representa la mejor solución operacional, ya que también deben considerarse factores como capacidad, cobertura, congestión, frecuencia y distribución espacial de la demanda.

3.4.4. Problemas de ruteo de vehículos

Los problemas de ruteo de vehículos estudian cómo organizar recorridos para atender un conjunto de puntos de demanda utilizando una flota disponible. A diferencia del problema de ruta mínima, en este caso no se busca únicamente conectar un origen con un destino, sino definir recorridos que visiten varios puntos de demanda bajo restricciones de capacidad, tiempo o número de vehículos. Dantzig y Ramser (1959) formularon uno de los primeros enfoques del problema de despacho de vehículos, mientras que Toth y Vigo (2002) desarrollan una revisión amplia sobre el problema de ruteo de vehículos y sus variantes.

Una formulación general simplificada del problema de ruteo de vehículos puede representarse como:

$$\min \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk}$$

sujeto a:

$$\sum_{k \in K} \sum_{j \in N} x_{ijk} = 1 \quad \forall i \in D$$

$$\sum_{i \in D} q_i \sum_{j \in N} x_{ijk} \leq Q_k \quad \forall k \in K$$

$$\sum_{j \in N} x_{hjk} - \sum_{i \in N} x_{ihk} = 0 \quad \forall h \in N, \forall k \in K$$

$$x_{ijk} \in \{0, 1\} \quad \forall i, j \in N, \forall k \in K$$

donde K representa el conjunto de vehículos, D el conjunto de puntos de demanda, q_i la demanda asociada al punto i , Q_k la capacidad del vehículo k , c_{ij} el costo de trasladarse desde i hacia j , y x_{ijk} una variable binaria que toma el valor de 1 si el vehículo k viaja

desde el nodo i hacia el nodo j .

En el caso de rutas de alimentación para Colina, este enfoque permite estudiar cómo recorrer distintos sectores de la comuna para captar pasajeros y conectarlos posteriormente con las rutas hacia Santiago. Su aplicación resulta útil cuando se requiere definir recorridos internos, asignar sectores a vehículos y controlar que la demanda atendida no supere la capacidad disponible.

3.4.5. Rutas troncales y rutas de alimentación

Las rutas troncales corresponden a recorridos de mayor jerarquía dentro de un sistema de transporte, generalmente asociados a mayores distancias, mayor capacidad o mayor concentración de demanda. En cambio, las rutas de alimentación cumplen la función de captar usuarios desde sectores específicos y conectarlos con una red principal. Esta organización permite diferenciar funciones dentro del sistema, separando los recorridos internos de menor escala de los desplazamientos principales hacia destinos de mayor jerarquía.

En términos de modelación, una ruta puede entenderse como una secuencia ordenada de nodos conectados por arcos de la red. Si se define una variable binaria x_{ij} , esta puede indicar si el arco (i, j) es utilizado dentro de una ruta:

$$x_{ij} = \begin{cases} 1, & \text{si el arco } (i, j) \text{ es utilizado en la ruta} \\ 0, & \text{en caso contrario} \end{cases}$$

Bajo esta lógica, el diseño de rutas busca seleccionar un conjunto de arcos que permita conectar zonas de demanda con puntos de acceso o corredores principales. En el caso de Colina, las rutas de alimentación no deben entenderse como recorridos destinados a concentrar toda la demanda en un único punto. Por el contrario, deben diseñarse como recorridos internos que atraviesen sectores relevantes de la comuna y luego se conecten con las rutas disponibles hacia Santiago. Esto permite mejorar la cobertura territorial sin generar cuellos de botella en un solo nodo de transferencia.

3.4.6. Restricciones de capacidad, tiempo y cobertura

El diseño de rutas de transporte debe considerar restricciones que representen las condiciones reales de operación. La capacidad limita la cantidad de pasajeros que puede transportar cada vehículo, el tiempo de recorrido incide en la calidad del servicio y en la posibilidad de cumplir frecuencias, mientras que la cobertura determina qué proporción de la demanda puede acceder al sistema bajo condiciones razonables.

Una restricción de capacidad puede expresarse como:

$$\sum_{i \in D} q_i z_{ik} \leq Q_k \quad \forall k \in K$$

donde z_{ik} toma el valor de 1 si la demanda del punto i es atendida por el vehículo k , q_i representa la demanda del punto i , y Q_k corresponde a la capacidad del vehículo.

Por otra parte, una restricción de tiempo máximo de recorrido puede formularse como:

$$\sum_{i \in N} \sum_{j \in N} t_{ij} x_{ijk} \leq T_k^{\max} \quad \forall k \in K$$

donde t_{ij} representa el tiempo de viaje entre los nodos i y j , y $T_k^{\max}$ corresponde al tiempo máximo permitido para la ruta del vehículo k .

Finalmente, una condición de cobertura puede establecer que un punto de demanda solo sea asignado a un punto de acceso si se encuentra dentro de una distancia o tiempo máximo aceptable:

$$z_{ij} \leq a_{ij} \quad \forall i \in D, \forall j \in P$$

donde a_{ij} toma el valor de 1 si el punto de demanda i se encuentra dentro del rango de cobertura del punto de acceso j , y 0 en caso contrario.

Estas restricciones son relevantes porque permiten evitar soluciones teóricas que no sean operativamente viables. En el caso de Colina, una ruta interna demasiado extensa

podría aumentar los tiempos de viaje y reducir la eficiencia del servicio, mientras que una ruta demasiado limitada podría dejar sectores sin cobertura. Por ello, el diseño debe equilibrar acceso, tiempo, capacidad y operación, considerando tanto la red interna de la comuna como su conexión con los corredores principales hacia Santiago.

3.4.7. OpenRouteService como herramienta de cálculo de rutas sobre la red vial

Los modelos de ruta mínima y de ruteo de vehículos requieren, como insumo, un costo o un tiempo asociado a cada arco de la red. En la práctica, este valor no se obtiene de forma teórica, sino que debe calcularse sobre una representación real de la infraestructura vial. Para ello se utiliza OpenRouteService (ORS), un servicio de ruteo de código abierto desarrollado por el grupo de investigación GIScience de la Universidad de Heidelberg (Neis & Zipf, 2008).

ORS opera sobre los datos cartográficos de OpenStreetMap (OSM), un proyecto colaborativo de cartografía libre que representa la red vial mundial mediante contribuciones abiertas y actualizables (Haklay & Weber, 2008). Con esta información, ORS construye un grafo dirigido cuyos nodos corresponden a intersecciones y cuyos arcos representan tramos de calle, cada uno con atributos de longitud, velocidad permitida y sentido de circulación. El motor de cálculo deriva de GraphHopper, un motor de ruteo de código abierto que resuelve el problema de ruta mínima mediante variantes del algoritmo de Dijkstra, aceleradas con técnicas de precómputo como las jerarquías de contracción (*contraction hierarchies*), las cuales permiten obtener rutas más cortas en tiempos de respuesta muy inferiores a los de una búsqueda exhaustiva sobre el grafo completo.

A través de su API, ORS entrega tres tipos de resultados de utilidad para el análisis de sistemas de transporte. El primero es una matriz de distancias y tiempos entre múltiples pares de nodos en una sola consulta. El segundo es la geometría de la ruta calculada entre dos puntos, útil para representar espacialmente los recorridos resultantes. El tercero es un servicio de ajuste de coordenadas a la red vial más cercana (*snapping*), empleado para corregir la posición de puntos georreferenciados que, por imprecisión en

su levantamiento, no caen exactamente sobre una vía transitable. Estos resultados pueden obtenerse bajo distintos perfiles de desplazamiento, como peatonal o vehicular, lo que permite adaptar el cálculo al tipo de acceso que se desea evaluar.

Frente a alternativas como Google Maps Platform, HERE o motores de ruteo similares basados en OpenStreetMap, como OSRM o Valhalla, ORS destaca por ser una herramienta de acceso abierto y gratuito, cuyos datos de origen son públicos y verificables, lo que favorece la reproducibilidad de los cálculos. No obstante, los tiempos y distancias obtenidos dependen directamente de la completitud y actualización de los datos de OpenStreetMap disponibles para el área de estudio.

En el eje Colina-Santiago, esta herramienta permite estimar de manera realista los tiempos de acceso peatonal hacia los puntos de recolección y los tiempos de viaje vehicular entre paraderos, accesos a autopista y los nodos de origen y destino, evitando que dichos costos se aproximen mediante distancias en línea recta que no representan las condiciones reales de la red vial de la comuna.

3.5. Caracterización del sistema de transporte Colina-Santiago

3.5.1. Contexto territorial y crecimiento urbano de Colina

Colina corresponde a una comuna ubicada en el sector norte de la Región Metropolitana, cuya estructura territorial combina áreas urbanas consolidadas, sectores rurales, zonas industriales y nuevos desarrollos residenciales. Esta condición genera una distribución espacial diversa de la población y de las actividades, lo que influye directamente en la forma en que se producen los desplazamientos diarios dentro y fuera de la comuna. Según el Censo 2017, la comuna registró 146.207 habitantes, cifra que representó un aumento de 87,89 % respecto de los 77.815 habitantes del Censo 2002. Además, la proyección 2024 estimó una población total de 197.370 habitantes, compuesta por 159.467 personas en áreas urbanas y 37.903 en áreas rurales (Biblioteca del Congreso Nacional de Chile, 2024).

Este crecimiento poblacional se ha visto acompañado por una expansión residencial relevante. En 2025, Colina registró el mayor número de ventas de casas nuevas de la Región Metropolitana, con 504 unidades comercializadas, equivalentes al 23 % del total regional informado por Colliers. Esta situación refuerza la tendencia de crecimiento hacia comunas periféricas, donde la disponibilidad de suelo ha impulsado nuevos desarrollos habitacionales (Chicureo Hoy, 2026). En consecuencia, el crecimiento urbano de Colina no solo aumenta la cantidad potencial de viajes, sino que también modifica su distribución espacial, generando nuevos sectores de origen y mayores exigencias sobre la infraestructura vial y el transporte público.

La localización de Colina también resulta relevante para comprender su dependencia funcional con Santiago. La distancia entre ambas zonas se ubica aproximadamente entre 33 y 39 kilómetros por ruta, dependiendo del punto específico de origen y destino considerado. Esta distancia permite viajes cotidianos hacia la capital, pero también implica una alta sensibilidad frente a congestión, tiempos de espera y disponibilidad de servicios de transporte. Por ello, el análisis del sistema comunal no puede limitarse a una única zona de origen, sino que debe considerar sectores como Colina centro y el sector de esmeralda, los cuales adquieren relevancia para estudiar la distribución de la demanda y la necesidad de puntos de acceso al transporte.

3.5.2. Dependencia funcional entre Colina y Santiago

La relación entre Colina y Santiago se explica por la dependencia funcional que existe entre una comuna periférica y el principal centro metropolitano del país. Una parte importante de los viajes cotidianos se origina en Colina y tiene como destino Santiago, debido a la concentración de empleos, servicios, educación superior, comercio y equipamientos especializados en la capital. Esta situación transforma al eje Colina-Santiago en un corredor relevante para la movilidad diaria de la población.

La dependencia funcional también implica que los problemas de transporte no se explican únicamente por la distancia física entre ambos territorios, sino por la capacidad del

sistema para conectar de manera eficiente los sectores internos de Colina con las rutas de salida hacia Santiago. Por esta razón, el análisis debe considerar dos niveles: la movilidad interna dentro de la comuna y la conexión interurbana hacia el área metropolitana.

3.5.3. Corredores principales de conexión

La conexión entre Colina y Santiago se estructura principalmente a través de corredores viales como Autopista Los Libertadores, Autopista Nororiental y carretera General San Martín. Estos ejes permiten canalizar los desplazamientos entre la comuna y la ciudad de Santiago, por lo que cumplen una función central dentro del sistema de movilidad interurbana. Su operación incide directamente en los tiempos de viaje, la congestión y la calidad del desplazamiento de los usuarios.

Sin embargo, la existencia de corredores principales no garantiza por sí sola un buen nivel de accesibilidad. Para que estos corredores funcionen adecuadamente, es necesario que los usuarios puedan acceder a ellos desde sus sectores de origen mediante recorridos internos, puntos de abordaje o servicios de acercamiento. En consecuencia, el problema no se reduce a la ruta hacia Santiago, sino a la forma en que la demanda comunal se organiza antes de incorporarse a dichos ejes.

3.5.4. Transporte público, transporte privado y patrones de movilidad

El sistema de movilidad de Colina presenta una participación relevante del transporte privado, especialmente en sectores donde la cobertura o frecuencia del transporte público resulta insuficiente. De acuerdo con un sondeo difundido por Chicureo Hoy (2022), un 63,9 % de los vecinos encuestados indicó movilizarse en vehículo particular. Este antecedente permite observar la importancia del automóvil dentro de la movilidad comunal y sugiere la existencia de una alta dependencia del transporte privado.

Esta situación puede generar efectos negativos sobre el sistema vial, especialmente en horarios punta, debido al aumento de vehículos particulares circulando hacia Santiago.

Además, quienes no disponen de automóvil enfrentan mayores restricciones de movilidad si la oferta de transporte público no logra cubrir adecuadamente sus necesidades. Por ello, resulta necesario estudiar alternativas que permitan mejorar el acceso al transporte colectivo y reducir la dependencia del vehículo particular.

4. Metodología

La metodología se desarrolla en la comuna de Colina, específicamente en los sectores residenciales que presentan una relación cotidiana de viajes hacia Santiago. El análisis se concentra en la identificación de nodos de demanda y puntos candidatos de recolección de pasajeros dentro del área urbana de la comuna, considerando principalmente sectores como Colina centro, Esmeralda y sus zonas residenciales cercanas. En este contexto, cada nodo de demanda representa una cuadra, conjunto habitacional o agrupación de viviendas desde donde podrían originarse viajes hacia Santiago, mientras que cada punto candidato representa una posible ubicación donde los usuarios podrían acceder caminando al sistema de transporte público.

La aplicación de esta metodología se justifica por la necesidad de mejorar la accesibilidad al transporte público interurbano y reducir la dependencia del automóvil privado. Actualmente, la baja cobertura, la incertidumbre del servicio y la lejanía de algunos puntos de abordaje obligan a muchos habitantes de Colina a salir con una anticipación excesiva para asegurar su llegada a Santiago. Por ejemplo, una persona que debe llegar a las 9:00 puede verse obligada a salir cerca de las 7:00 debido a la falta de alternativas cercanas o confiables. Al acercar los puntos de recolección de pasajeros a los sectores residenciales, se busca disminuir los tiempos muertos, mejorar la certidumbre del viaje y favorecer una distribución más equilibrada de la demanda durante las horas punta. De esta manera, la propuesta también podría contribuir a aliviar la presión sobre las carreteras y accesos principales hacia Santiago, al ofrecer una alternativa más cercana y competitiva frente al automóvil privado.

El desarrollo metodológico se realiza mediante un enfoque de localización y cobertura. En primer lugar, se construye una base de datos georreferenciada en formato KMZ, donde se identifican nodos de demanda y puntos candidatos de recolección de pasajeros. Luego, esta información se procesa mediante herramientas computacionales para extraer coordenadas, ordenar los datos y construir matrices de distancia y tiempo de caminata entre cada nodo de demanda y cada punto candidato. A partir de estas matrices se formula el modelo de Cobertura Máxima (Church y ReVelle, 1974), cuyo objetivo es seleccionar un subconjunto de puntos candidatos que maximice la demanda cubierta bajo un estándar de accesibilidad óptima, asegurando, al mismo tiempo, que todos los nodos de demanda considerados queden dentro de un radio de cobertura aceptable.

De esta forma, la metodología transforma la información territorial en una estructura cuantitativa de decisión. Cada sector residencial se representa mediante un nodo de demanda, cada posible ubicación para la recolección de pasajeros se representa mediante un nodo candidato, y la relación entre ambos se define a partir de distancias y tiempos de caminata sobre la red vial. Esto permite que la selección de puntos no dependa únicamente de una observación visual del territorio, sino de un criterio sistemático de cobertura, accesibilidad y localización.

4.1. Formulación del modelo de localización

El problema que se aborda en este trabajo corresponde a una variante del modelo de Cobertura Máxima (Church y ReVelle, 1974), que busca seleccionar un subconjunto de puntos dentro de un conjunto de candidatos, de manera de maximizar la demanda cubierta. El modelo incorpora dos radios de servicio: un primer radio R_1 , asociado a una cobertura óptima, y un segundo radio R_2 , asociado a una cobertura deseable, menos exigente que la anterior. La demanda cubierta dentro de cada radio se incorpora en la función objetivo de forma ponderada, priorizando la cobertura dentro de R_1 por sobre la cobertura adicional que se obtiene únicamente dentro de R_2 .

Para cada nodo de demanda i se definen vecindades $N_1(i)$ y $N_2(i)$, que contienen los puntos candidatos ubicados dentro de los radios R_1 y R_2 , respectivamente. La decisión principal consiste en determinar qué puntos de recolección de pasajeros se seleccionan y cómo se asigna la demanda hacia dichos puntos, sujeto a una cantidad fija de puntos a habilitar y a restricciones de capacidad por punto.

Los radios R_1 y R_2 se definen a partir del mismo estándar del Sistema de Indicadores y Estándares de Desarrollo Urbano (SIEDU) del Consejo Nacional de Desarrollo Urbano. R_1 corresponde al valor de referencia establecido para el acceso a paraderos de transporte público (Instituto Nacional de Estadísticas, 2019, Indicador BPU_25), mientras que R_2 corresponde al área de influencia de un paradero de transporte público mayor (Instituto Nacional de Estadísticas, 2020; Consejo Nacional de Desarrollo Urbano, 2018). El INE establece como deseable que, dentro de esta área de influencia, quede cubierto al menos el 90 % de la población; este valor se utiliza en el presente trabajo como referencia de cobertura que se espera alcanzar con el modelo propuesto, evaluándose en cada escenario qué tan cerca se llega a ese resultado.

A continuación se presenta el modelo propuesto, utilizando las notaciones definidas para conjuntos, parámetros, variables de decisión, función objetivo y restricciones.

Conjuntos

I : Conjunto de nodos de demanda, $i \in I$.

J : Conjunto de puntos candidatos de recolección de pasajeros, $j \in J$.

$N_1(i) = \{j \in J : d_{ij} \leq R_1\}$: Conjunto de puntos candidatos que cumplen el estándar de cobertura óptima para el nodo de demanda i , $\forall i \in I$.

$N_2(i) = \{j \in J : d_{ij} \leq R_2\}$: Conjunto de puntos candidatos que cumplen el estándar de cobertura deseable para el nodo de demanda i , $\forall i \in I$.

Parámetros

h_i : Demanda estimada asociada al nodo de demanda i , $\forall i \in I$.

d_{ij} : Distancia de caminata entre el nodo de demanda i y el punto candidato j , $\forall i \in I, j \in J$.

R_1 : Radio máximo asociado al estándar de cobertura óptimo.

R_2 : Radio máximo asociado al estándar de cobertura deseable.

p : Número exacto de puntos de recolección de pasajeros que se deben habilitar.

b_j : Capacidad máxima de atención del punto candidato j , expresada como la cantidad máxima de usuarios que puede recibir en el periodo de análisis, $\forall j \in J$.

O_1 : Peso asociado a la cobertura óptima. ($O_1 = 2.0$)

Variables de decisión

$$X_j = \begin{cases} 1, & \text{si se selecciona el punto candidato } j \\ 0, & \text{en otro caso} \end{cases} \quad \forall j \in J$$

donde X_j indica si el punto candidato j es seleccionado como punto de recolección de pasajeros.

$$Z_i = \begin{cases} 1, & \text{si el nodo de demanda } i \text{ queda cubierto dentro de } R_1 \\ 0, & \text{en otro caso} \end{cases} \quad \forall i \in I$$

donde Z_i indica si el nodo de demanda i queda cubierto dentro del radio de cobertura óptima.

$$W_i = \begin{cases} 1, & \text{si el nodo de demanda } i \text{ queda cubierto dentro de } R_2 \\ 0, & \text{en otro caso} \end{cases} \quad \forall i \in I$$

donde W_i indica si el nodo de demanda i queda cubierto dentro del radio de cobertura deseable.

$$Y_{ij} = \begin{cases} 1, & \text{si la población } i \text{ está cubierta por un punto de recolección } j \\ 0, & \text{en otro caso} \end{cases} \quad \forall i \in I, j \in J$$

donde Y_{ij} indica si el nodo de demanda i es asignado al punto de recolección j .

Función objetivo

Maximizar la demanda cubierta dentro del estándar de cobertura óptima:

$$\text{máx } F = O_1 \sum_{i \in I} h_i Z_i + \sum_{i \in I} h_i W_i$$

Esta función busca maximizar la cantidad total de usuarios potenciales cubiertos por los puntos de recolección seleccionados. El primer término prioriza la demanda que queda dentro del radio óptimo R_1 , mientras que el segundo término incorpora la demanda que queda cubierta dentro del radio deseable R_2 . De esta manera, el modelo favorece soluciones que acercan el sistema de transporte público a los sectores residenciales, sin excluir la cobertura de sectores que, aun no estando dentro del radio óptimo, sí se encuentran dentro de una distancia caminable aceptable.

Restricciones

(1) Cobertura óptima de la demanda:

$$Z_i \leq \sum_{j \in N_1(i)} Y_{ij} \quad \forall i \in I$$

Esta restricción establece que un nodo de demanda i solo puede considerarse cubierto dentro del radio óptimo R_1 si es asignado a un punto de recolección ubicado

dentro de su vecindad $N_1(i)$.

(2) Cobertura deseable de la demanda:

$$\sum_{j \in N_2(i)} Y_{ij} = W_i \quad \forall i \in I$$

Esta restricción indica que si un nodo de demanda queda cubierto dentro del radio deseable R_2 , entonces debe ser asignado a un único punto de recolección dentro de dicho radio. Si el nodo no queda cubierto, no se asigna a ningún punto.

(3) Relación entre cobertura óptima y cobertura deseable:

$$Z_i \leq W_i \quad \forall i \in I$$

Esta restricción asegura que todo nodo cubierto dentro del radio óptimo R_1 también sea considerado cubierto dentro del radio deseable R_2 .

(4) Vinculación entre asignación y apertura:

$$Y_{ij} \leq X_j \quad \forall i \in I, j \in N_2(i)$$

Esta restricción evita asignar demanda a puntos candidatos que no hayan sido seleccionados.

(5) Capacidad máxima de los puntos de recolección:

$$\sum_{i \in I: j \in N_2(i)} h_i Y_{ij} \leq b_j X_j \quad \forall j \in J$$

Esta restricción asegura que la demanda asignada a cada punto de recolección no supere su capacidad máxima de atención. Si el punto candidato j no es seleccionado, entonces ($X_j = 0$) y no puede recibir demanda.

(6) Número de puntos de recolección a seleccionar:

$$\sum_{j \in J} X_j = p$$

Esta restricción garantiza que se seleccionen exactamente p puntos de recolección de pasajeros.

(7) Dominios de las variables:

$$X_j \in \{0, 1\} \quad \forall j \in J$$

$$Z_i \in \{0, 1\} \quad W_i \in \{0, 1\} \quad \forall i \in I$$

$$Y_{ij} \in \{0, 1\} \quad \forall i \in I, j \in N_2(i)$$

Interpretación del modelo

El modelo propuesto permite seleccionar los puntos de recolección de pasajeros que entregan una mayor cobertura de la demanda dentro del radio óptimo R_1 , considerando una cantidad fija de puntos a habilitar. Adicionalmente, se evalúa la cobertura alcanzada dentro del radio deseable R_2 respecto del estándar de referencia del 90 % de la demanda total, utilizado como parámetro de comparación para el nivel de accesibilidad logrado en el área de estudio.

La variable X_j identifica los puntos candidatos que son seleccionados como puntos de recolección. La variable Z_i indica si el nodo de demanda i queda cubierto dentro del radio óptimo, mientras que la variable W_i indica si dicho nodo queda cubierto dentro del radio deseable. Por su parte, la variable Y_{ij} asigna cada nodo de demanda cubierto a un punto de recolección seleccionado, lo que permite incorporar la capacidad máxima de atención de cada punto.

Desde el punto de vista del sistema de transporte público Colina-Santiago, esta formulación permite construir una red inicial de puntos de recolección cercana a los sectores residenciales, considerando criterios de cobertura, capacidad y accesibilidad peatonal. De esta manera, el modelo entrega una herramienta de apoyo para definir ubicaciones que

faciliten el acceso al transporte público, reduzcan la dependencia del automóvil privado y contribuyan a una organización más eficiente de los viajes en los períodos de mayor demanda.

4.1.1. Análisis del problema y estrategia de resolución

El problema planteado corresponde a un modelo de localización discreta de puntos de recolección de pasajeros para el sistema de transporte público interurbano entre Colina y Santiago. Su objetivo es seleccionar un subconjunto de puntos candidatos que permita maximizar la demanda cubierta dentro de un radio de cobertura óptima, evaluando adicionalmente la cobertura alcanzada dentro de un radio de cobertura deseable respecto de un estándar de referencia del 90 % de la demanda total. La formulación considera un conjunto de nodos de demanda, representados por cuadras, conjuntos habitacionales o agrupaciones residenciales, y un conjunto de puntos candidatos, correspondientes a posibles ubicaciones donde los usuarios podrían acceder caminando al sistema de transporte público.

La estructura del modelo combina decisiones binarias de selección de puntos candidatos con decisiones de cobertura y asignación de demanda. La variable de apertura determina qué puntos candidatos son seleccionados como puntos de recolección, mientras que las variables de cobertura identifican qué nodos de demanda quedan cubiertos dentro de los radios definidos. La variable de asignación, por su parte, asocia cada nodo cubierto a un punto de recolección específico, incorporando restricciones de capacidad máxima de atención en cada punto seleccionado.

El modelo busca equilibrar accesibilidad territorial, cobertura de referencia y factibilidad operacional. Por una parte, se maximiza la demanda cubierta dentro del radio óptimo R_1 , priorizando que la mayor cantidad posible de usuarios se encuentre a una distancia caminable adecuada del sistema. Por otra parte, se evalúa la cobertura alcanzada dentro del radio deseable R_2 respecto de un estándar de referencia del 90 % de la demanda total, como indicador del nivel de accesibilidad del sistema propuesto.

Finalmente, la restricción de capacidad evita concentrar una cantidad excesiva de usuarios en un único punto de recolección, permitiendo que la demanda sea asignada a otros puntos seleccionados cuando exista capacidad disponible.

La resolución del problema se realiza mediante un enfoque de optimización exacta. Para ello, la formulación se expresa como un modelo de programación lineal entera mixta, debido a la presencia de variables binarias asociadas a la selección de puntos, cobertura y asignación. El modelo será implementado en Python mediante la librería Pyomo, lo que permite construir de forma ordenada los conjuntos, parámetros, variables, función objetivo y restricciones.

Para resolver el modelo se utilizará un solver de programación matemática, como GLPK o CBC. Estos solvers permiten resolver problemas lineales enteros mediante algoritmos de ramificación y acotamiento, los cuales exploran distintas combinaciones de variables binarias hasta encontrar una solución óptima o una solución factible de buena calidad. En cada iteración, el solver resuelve relajaciones lineales del problema y descarta ramas del árbol de búsqueda que no pueden mejorar la solución encontrada, reduciendo así el espacio de combinaciones posibles.

La elección de este enfoque se justifica porque el problema presenta una estructura discreta y combinatoria, donde se debe decidir qué puntos candidatos seleccionar dentro de un conjunto finito de alternativas. Además, el tamaño del problema es manejable para una implementación computacional, considerando que la base georreferenciada contiene un número definido de nodos de demanda y puntos candidatos. Esto permite obtener una solución reproducible y trazable, manteniendo coherencia entre la información territorial levantada y los resultados del modelo.

El procedimiento de resolución se desarrolla de acuerdo con las siguientes etapas:

1. Construcción de la base de datos georreferenciada, identificando nodos de demanda y puntos candidatos de recolección de pasajeros en la comuna de Colina.
2. Procesamiento de las coordenadas de cada nodo mediante Python, separando la información en tablas de nodos de demanda y puntos candidatos, para luego construir

la matriz de distancias y tiempos de caminata entre cada nodo de demanda i y cada punto candidato j . A partir de esta matriz, se definen las vecindades $N_1(i)$ y $N_2(i)$, que agrupan los puntos candidatos ubicados dentro del radio óptimo R_1 y del radio deseable R_2 , respectivamente.

3. Generación de los parámetros del modelo, incluyendo la demanda estimada h_i , la capacidad máxima b_j , los radios de cobertura, el peso asociado a la cobertura óptima O_1 y el número de puntos a seleccionar p .
4. Formulación y resolución del modelo matemático en Pyomo mediante un solver de programación lineal entera mixta, incorporando la función objetivo de maximización de cobertura y las restricciones de cobertura mínima, asignación, capacidad y selección de puntos, para obtener los puntos de recolección seleccionados, los nodos cubiertos y la asignación de demanda.
5. Análisis de resultados, considerando indicadores como demanda cubierta dentro del radio óptimo, demanda cubierta dentro del radio deseable, porcentaje de cobertura total, utilización de capacidad de los puntos seleccionados y distribución espacial de la cobertura.

Esta estrategia permite transformar la información georreferenciada en una herramienta cuantitativa de apoyo a la toma de decisiones. A partir de la solución obtenida, es posible identificar qué puntos de recolección presentan mayor conveniencia territorial, qué sectores quedan cubiertos dentro de los radios definidos y si la red propuesta cumple con el porcentaje mínimo de cobertura requerido. Además, la formulación permite evaluar escenarios posteriores modificando el número de puntos seleccionados, la capacidad máxima de los puntos de recolección o los radios de cobertura, sin alterar la estructura general del modelo.

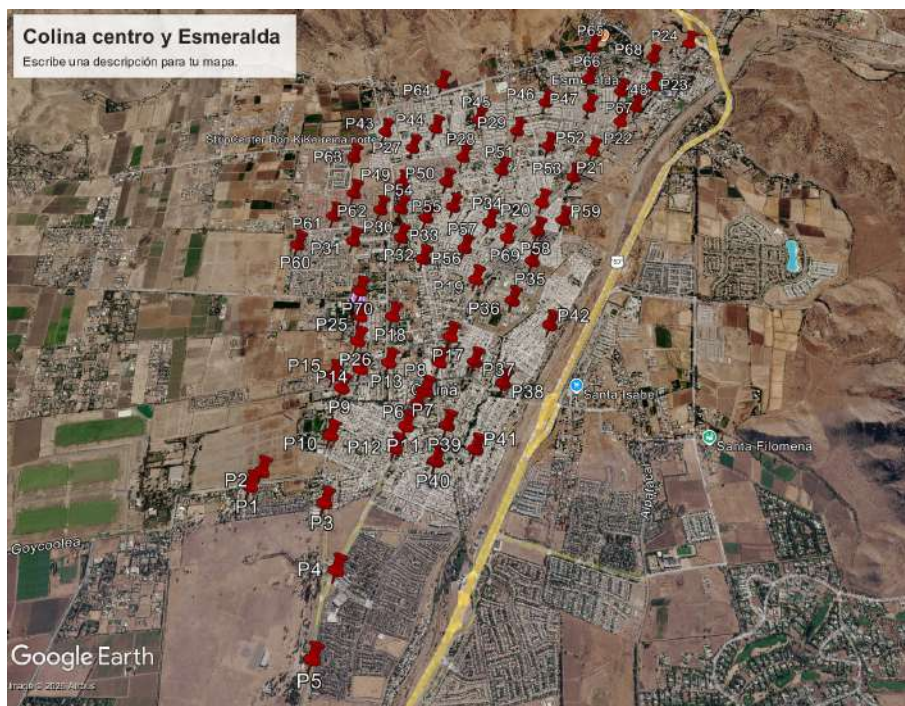
4.1.2. Construcción de la base de datos georreferenciada

La primera etapa consiste en la construcción de una base de datos georreferenciada para la comuna de Colina. Para ello se utiliza Google Earth Pro como herramienta principal de levantamiento espacial, complementada con la observación directa de la estructura urbana, la red vial y la distribución de las zonas residenciales.

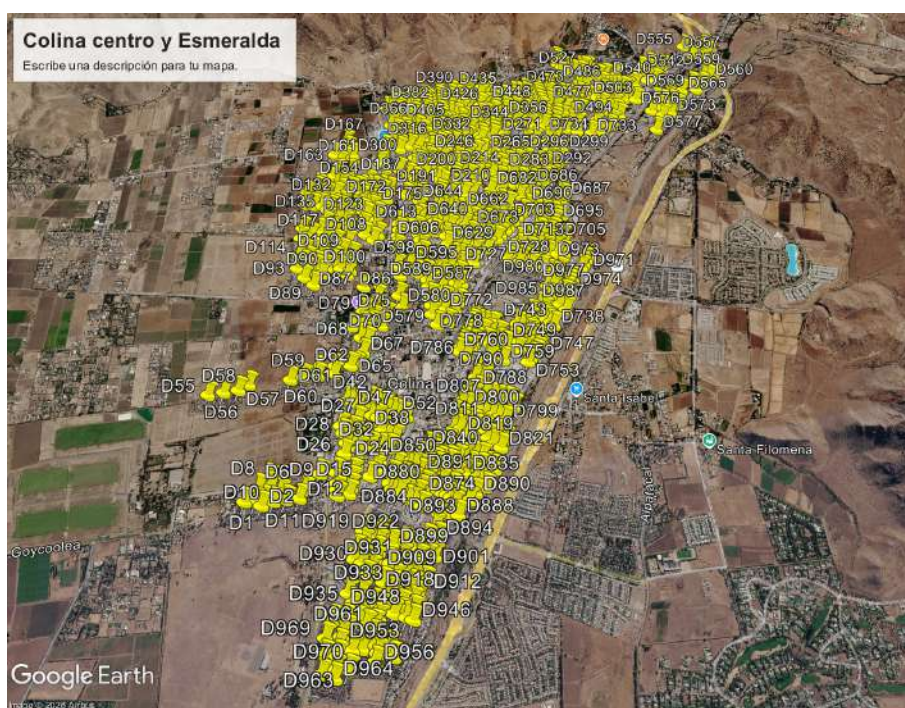
Los nodos de demanda se definieron mediante la técnica de agregación de demanda por centroides, ampliamente utilizada en modelos de localización de cobertura para representar la demanda continua de un territorio mediante un conjunto finito de puntos representativos (Current y Schilling, 1990). Bajo este enfoque, cada nodo se ubica en el centroide aproximado de una cuadra, conjunto habitacional o agrupación residencial identificable en la cartografía, considerando el amanzanamiento propio de cada sector. La ubicación de los nodos evita terrenos vacíos, áreas industriales sin generación residencial relevante y duplicación innecesaria de puntos dentro de una misma agrupación de viviendas. Cada nodo de demanda queda registrado con una identificación única y coordenadas geográficas.

De manera paralela, se identifican puntos candidatos para funcionar como puntos de recolección de pasajeros. Estos corresponden a lugares donde los usuarios podrían llegar caminando desde sus viviendas para abordar un bus o servicio de transporte público. Los puntos candidatos pueden ubicarse en paraderos existentes, accesos principales a barrios, intersecciones relevantes, equipamientos públicos, plazas o sectores con buena conectividad hacia la red vial principal. Su ubicación considera además la separación entre paraderos de locomoción colectiva establecida en la Ordenanza General de Urbanismo y Construcciones (OGUC), Artículo 2.3.2, que fija un distanciamiento preferente superior a 300 metros en vías de servicio y superior a 500 metros en vías troncales, lo que explica que en algunos sectores los puntos candidatos queden relativamente cercanos entre sí. Cada punto candidato se registra como una alternativa potencial dentro del modelo de localización.

El resultado de esta etapa es una base cartográfica en formato KMZ, compuesta



Imágen 2: Puntos candidatos de recolección de pasajeros en el sector Colina centro y Esmeralda.



Imágen 3: Nodos de demanda representados en el sector Colina centro y Esmeralda.

En total, la base georreferenciada construida para el sector Colina centro y Esmeralda considera 70 puntos candidatos de recolección de pasajeros y 991 nodos de demanda.

Los puntos candidatos fueron identificados en color rojo y corresponden a ubicaciones potenciales donde los usuarios podrían abordar el sistema de transporte público. Por su parte, los nodos de demanda fueron identificados en color amarillo y representan centroides aproximados de cuadras o agrupaciones residenciales. Esta cantidad de puntos permite representar con mayor detalle la distribución espacial de la demanda dentro del área de estudio y sirve como base para la posterior formulación del modelo de localización.

4.1.3. Procesamiento espacial y construcción de vecindades

A partir de la base georreferenciada construida en Google Earth Pro, la información espacial se procesa mediante Python para transformar los puntos levantados en insumos utilizables por el modelo de localización. En esta etapa se importa el archivo geográfico y se separan las capas correspondientes a nodos de demanda y puntos candidatos de recolección. Luego, se extraen las coordenadas de latitud y longitud de cada elemento, asignando un identificador único a cada nodo y a cada punto candidato.

Con las coordenadas procesadas se construye la matriz de distancias d_{ij} , donde cada valor representa la distancia de caminata entre el nodo de demanda i y el punto candidato j . El cálculo se realiza mediante OpenRouteService, utilizando el perfil *foot-walking*, con el propósito de representar recorridos peatonales sobre la red disponible y no distancias lineales entre coordenadas. De forma complementaria, se obtiene una matriz de tiempos de caminata expresada en minutos.

A partir de la matriz de distancias d_{ij} se definen las vecindades $N_1(i)$ y $N_2(i)$. La vecindad $N_1(i)$ contiene los puntos candidatos que se encuentran dentro del radio de cobertura óptima R_1 para cada nodo de demanda, mientras que la vecindad $N_2(i)$ contiene los puntos candidatos ubicados dentro del radio de cobertura deseable R_2 . Estas vecindades permiten limitar las posibles asignaciones de demanda únicamente a puntos de recolección que se encuentren dentro de una distancia caminable considerada válida por el modelo.

El procedimiento computacional utilizado para importar la información georrefer-

renciada, extraer las coordenadas, calcular las matrices de distancia y tiempo, y construir los insumos del modelo se presenta en el Anexo 8.1.

4.1.4. Generación de parámetros del modelo

La generación de parámetros del modelo se realiza a partir de la base georreferenciada, de la matriz de distancias caminables y de antecedentes técnicos asociados a los servicios de buses rurales que operan en la Región Metropolitana. En particular, se utilizan como referencia los antecedentes del estudio *Análisis de demanda de pasajeros en servicios rurales de la Región Metropolitana*, elaborado para la Subsecretaría de Transportes, y el Oficio N° 16694/2026 de la Subsecretaría de Transportes, donde se indican fuentes oficiales de información sobre demanda, validaciones, contadores de pasajeros, trazados, frecuencias y operación de servicios vinculados al eje Colina–Santiago (Aristo Consultores, 2022; Subsecretaría de Transportes, 2026).

El parámetro d_{ij} corresponde a la distancia de caminata entre el nodo de demanda i y el punto candidato de recolección j . Este valor se obtiene directamente desde la matriz de distancias generada mediante OpenRouteService, utilizando el perfil *foot-walking*. Por lo tanto, d_{ij} no se define de forma manual, sino que se calcula a partir de la red caminable disponible entre cada par de nodos. Esta decisión es consistente con los resultados del estudio de buses rurales, donde se observa que el modo de acceso predominante al bus rural es la caminata. En dicho estudio, cerca del 92 % de los pasajeros declara llegar caminando al lugar donde sube al bus, lo que respalda el uso de distancias peatonales para evaluar la accesibilidad hacia los puntos de recolección (Aristo Consultores, 2022).

Los radios de cobertura se definen en kilómetros, dado que la matriz d_{ij} se encuentra expresada en dicha unidad. Para el escenario base se establece un radio de cobertura óptima de $R_1 = 0,4$ km y un radio de cobertura deseable de $R_2 = 0,5$ km. El valor de R_1 se asocia al estándar de 400 metros de distancia a paraderos de transporte público mayor utilizado por el SIEDU, mientras que R_2 se utiliza como un umbral de cobertura deseable, más amplio que el radio óptimo, para representar una distancia caminable aceptable dentro

del modelo. El INE establece como deseable que, dentro de esta área de influencia, quede cubierto al menos el 90 % de la población; este valor se utiliza en el presente trabajo como referencia de cobertura que se espera alcanzar con el modelo propuesto, evaluándose en cada escenario qué tan cerca se llega a ese resultado.

La demanda asociada a cada nodo se representa mediante el parámetro h_i . Para su estimación se considera que la demanda total del área de estudio está compuesta por una demanda actual observada y una demanda potencial:

$$H_C = H_{\text{actual}} + H_{\text{potencial}}$$

donde H_{actual} corresponde a la demanda observada en los servicios actuales de buses rurales asociados a Colina, mientras que $H_{\text{potencial}}$ representa una demanda adicional que podría incorporarse al sistema producto de la mejora de accesibilidad generada por los nuevos puntos de recolección de pasajeros.

La demanda actual se obtiene a partir de la base de mediciones de demanda de buses rurales, filtrando los servicios asociados a Colina y el escenario de análisis definido. Para el escenario base se considera un día laboral, el periodo punta mañana y el sentido ida, dado que representa el flujo principal de viajes cotidianos desde Colina hacia Santiago. Este periodo comprende las horas 6 a 10, es decir, una ventana de 5 horas. En consecuencia, la demanda utilizada en el modelo no representa la demanda diaria total, sino la cantidad de pasajeros estimada para dicha ventana horaria y sentido de viaje. Por lo tanto, los parámetros H_{actual} , H_C y h_i quedan expresados en pasajeros por periodo de análisis de 5 horas.

Esta base resulta pertinente, ya que el estudio registra subidas y bajadas de pasajeros, características de los usuarios, tarifa pagada y coordenadas asociadas a los lugares donde se produce la transferencia de pasajeros. Además, las mediciones son amplificadas mediante factores de expansión, lo que permite estimar el volumen total de pasajeros transportados por servicio, sentido, hora y tipo de día (Aristo Consultores, 2022).

Para incorporar la demanda potencial, se define un porcentaje de aumento sobre la

demanda actual observada:

$$H_{\text{potencial}} = \gamma H_{\text{actual}}$$

donde γ corresponde al factor de aumento de demanda potencial. De esta forma, la demanda total del escenario queda expresada como:

$$H_C = H_{\text{actual}}(1 + \gamma)$$

Para el escenario base se considera un factor de aumento de demanda potencial de $\gamma = 0,50$, equivalente a un incremento del 50 % respecto de la demanda actual observada. Considerando que la demanda actual filtrada para Colina en día laboral, punta mañana y sentido ida corresponde a $H_{\text{actual}} = 1,973,33$ pasajeros, se obtiene:

$$H_C = 1,973,33(1 + 0,50) = 2,960,00$$

Una vez definida la demanda total H_C , esta se distribuye entre los nodos de demanda mediante una distribución uniforme ajustada al número total de nodos. Dado que el modelo considera $|I|$ nodos de demanda, la demanda promedio esperada por nodo corresponde a $H_C/|I|$. Por ello, la demanda asociada a cada nodo se genera a partir de la siguiente distribución:

$$h_i \sim U\left(0, \frac{2H_C}{|I|}\right) \quad \forall i \in I$$

Esta distribución permite incorporar variabilidad entre los nodos de demanda, evitando asumir que todas las cuadras o agrupaciones residenciales generan exactamente la misma cantidad de viajes. Además, el valor esperado de la demanda de cada nodo queda dado por:

$$E(h_i) = \frac{H_C}{|I|}$$

por lo que la suma esperada de la demanda de todos los nodos corresponde a la demanda total estimada del área de estudio:

$$E \left(\sum_{i \in I} h_i \right) = H_C$$

La demanda no pudo estimarse a partir de mediciones reales desagregadas por nodo, ya que ese nivel de detalle solo está disponible para los servicios de transporte público licitado, cuyos operadores están contractualmente obligados a reportar información de subidas y bajadas de pasajeros como parte de sus condiciones de concesión. El transporte privado remunerado de pasajeros (Decreto Supremo N°80 de 2004, Ministerio de Transportes y Telecomunicaciones), es decir, colectivos, taxis y transporte escolar, no está sujeto a una obligación equivalente, por lo que no existe un registro público de demanda a nivel de nodo para estos modos. Por este motivo, la demanda se genera mediante la distribución uniforme descrita anteriormente.

Para el escenario base, considerando una demanda total ajustada de $H_C = 2,960,00$ pasajeros y un total de $|I| = 991$ nodos de demanda, la demanda promedio esperada por nodo es:

$$\frac{H_C}{|I|} = \frac{2,960,00}{991} = 2,99$$

Por lo tanto, la demanda asociada a cada nodo se genera según:

$$h_i \sim U(0, 5,98) \quad \forall i \in I$$

El parámetro b_j representa la capacidad máxima de atención del punto candidato j durante el periodo de análisis. A diferencia de h_i , este parámetro no se genera de forma aleatoria: se define como un valor fijo, idéntico para todos los puntos candidatos y para los seis escenarios evaluados, calculado a partir de referencias reales de capacidad vehicular y frecuencia de servicio:

$$b_j = C_{\text{bus}} \cdot f \cdot T \quad \forall j \in J$$

donde C_{bus} corresponde a la capacidad de un bus estándar del sistema RED Metropolitana de Movilidad (28 pasajeros sentados y 63 de pie, equivalente a 90 pasajeros), cuya base regulatoria es el Decreto Supremo N°122 de 1991 del Ministerio de Transportes y Telecomunicaciones; f corresponde a una frecuencia de servicio asumida de 2 buses por hora, equivalente a un bus cada 30 minutos, consistente con un servicio alimentador de baja demanda; y T corresponde a la duración del periodo de análisis, de 5 horas, verificada a partir de la base de demanda real utilizada para estimar H_{actual} . Con estos valores:

$$b_j = 90 \cdot 2 \cdot 5 = 900 \text{ pasajeros por periodo} \quad \forall j \in J$$

El parámetro p indica el número de puntos de recolección de pasajeros que se deben habilitar. En lugar de utilizar un único valor fijo, este parámetro se evalúa mediante distintos escenarios, considerando $p = [20, 30, 40, 50, 60, 70]$. Esta comparación permite analizar cómo cambia la cobertura de la demanda al aumentar la cantidad de puntos seleccionados.

El propósito de evaluar distintos valores de p es identificar hasta qué punto la incorporación de nuevos puntos de recolección genera mejoras relevantes en la cobertura. De esta forma, se puede observar si existe un comportamiento de rendimientos decrecientes, es decir, si al aumentar el número de puntos seleccionados la cobertura mejora cada vez menos. Esto permite seleccionar una alternativa que equilibre cobertura territorial y cantidad de puntos habilitados, evitando sobredimensionar la red de recolección.

Finalmente, el parámetro O_1 corresponde al peso asociado a la cobertura óptima. Para el escenario base se define $O_1 = 2$, lo que implica que cubrir demanda dentro del radio óptimo R_1 tiene el doble de valoración que cubrirla únicamente dentro del radio deseable R_2 . Esta ponderación permite que el modelo no solo busque cubrir demanda, sino que priorice soluciones en que los usuarios se encuentren más cerca de los puntos de

recolección seleccionados. La Tabla 1 resume los principales parámetros utilizados en el modelo.

Tabla 1: Parámetros base utilizados en el modelo de localización

Parámetro	Descripción	Valor base
d_{ij}	Distancia de caminata entre demanda i y candidato j	Matriz ORS
R_1	Radio de cobertura óptima	0,4 km
R_2	Radio de cobertura deseable	0,5 km
H_{actual}	Demanda actual observada en servicios de Colina (5 h)	1,973,33
$H_{\text{potencial}}$	Demanda adicional por mejora de accesibilidad	γH_{actual}
γ	Factor de aumento por demanda potencial	0,50
H_C	Demanda total estimada del área de estudio	2,960,00
h_i	Demanda estimada del nodo i	$U(0, 5,98)$
C_{bus}	Capacidad de un bus estándar (RED Metropolitana)	90 pasajeros
f	Frecuencia de servicio asumida	2 buses/hora
T	Duración del periodo de análisis	5 horas
b_j	Capacidad máxima del punto candidato j (fija)	900
p	Número de puntos de recolección a seleccionar	20, 30, 40, 50, 60, 70
O_1	Peso asociado a cobertura óptima	2

4.1.5. Implementación y resolución del modelo en Pyomo

La implementación computacional del modelo se realiza en Python mediante la librería Pyomo. En primer lugar, el código importa un archivo en formato JSON, el cual contiene los nodos de demanda, los puntos candidatos de recolección y la matriz de distancias caminables calculada previamente. A partir de esta información, se ordenan los identificadores de demanda y candidatos, y se construye la matriz d_{ij} que será utilizada por el modelo.

Luego, el código genera los parámetros necesarios para la resolución. La demanda de cada nodo h_i se obtiene mediante una distribución uniforme ajustada a la demanda total

estimada H_C , mientras que la capacidad máxima de cada punto candidato b_j se genera para cada escenario en función del valor de p . De esta forma, los parámetros utilizados por el modelo quedan definidos de manera automática antes de iniciar la optimización.

Posteriormente, se construyen las vecindades de cobertura $N_1(i)$ y $N_2(i)$ a partir de la matriz de distancias. Para cada nodo de demanda, el código identifica qué puntos candidatos se encuentran dentro del radio óptimo R_1 y cuáles se encuentran dentro del radio deseable R_2 . Estas vecindades permiten reducir el número de combinaciones evaluadas, ya que solo se crean variables de asignación Y_{ij} para pares de nodos y candidatos que se encuentran dentro del radio deseable.

Una vez preparados los insumos, se define el modelo en Pyomo mediante `ConcreteModel()`. En esta etapa se declaran los conjuntos, parámetros y variables binarias del modelo. Luego, se incorporan las restricciones de cobertura, asignación, vinculación entre asignación y apertura, capacidad máxima y selección de puntos. Finalmente, se define la función objetivo de maximización y se resuelve el modelo mediante un solver de programación lineal entera mixta.

La resolución se realiza para distintos valores del parámetro p , considerando los escenarios $p = 20$, $p = 30$, $p = 40$, $p = 50$, $p = 60$ y $p = 70$. Para cada escenario, el código entrega los principales resultados del modelo, incluyendo el valor de la función objetivo, el porcentaje de demanda cubierta dentro de R_1 y R_2 , la cantidad de nodos cubiertos y los puntos de recolección seleccionados. Estos resultados se utilizan posteriormente para comparar los escenarios y representar espacialmente la solución obtenida.

4.1.6. Análisis de resultados

Una vez resuelto el modelo, se comparan los escenarios definidos para el parámetro p con el objetivo de evaluar el efecto que tiene aumentar la cantidad de puntos de recolección seleccionados. Para ello, se consideran indicadores como la demanda cubierta dentro del radio óptimo R_1 , la demanda cubierta dentro del radio deseable R_2 , el porcentaje de cobertura alcanzado y la cantidad de nodos cubiertos en cada caso.

El análisis permite identificar el escenario que entrega una mejor relación entre cobertura y número de puntos habilitados. En particular, se observa si el aumento de puntos genera mejoras significativas o si, por el contrario, la cobertura comienza a estabilizarse, evidenciando rendimientos decrecientes. De esta forma, los resultados permiten seleccionar una alternativa que mejore la accesibilidad sin sobredimensionar la red de puntos de recolección.

4.2. Formulación del modelo de ruteo

Una vez obtenidos los puntos de recolección mediante el modelo de localización, se plantea una segunda etapa orientada a la construcción de rutas alimentadoras teóricas. Esta etapa no decide dónde ubicar los paraderos, ya que estos fueron determinados previamente, sino que define cómo conectarlos mediante recorridos que comienzan en un nodo de origen, visitan un subconjunto de paraderos, utilizan un acceso a autopista y finalizan en un nodo de destino asociado a la conexión hacia Santiago.

El modelo formulado considera el sentido de viaje Colina-Santiago, correspondiente al flujo principal de demanda en el período punta mañana. El recorrido de regreso (Santiago-Colina) se asume equivalente a la misma ruta, recorrida en sentido inverso, por lo que no requiere una optimización independiente para dicho sentido.

El modelo de ruteo corresponde a una variante del problema de ruteo de vehículos (VRP), en la que la duración máxima de cada ruta se limita mediante un tiempo total permitido, en lugar de una restricción de capacidad de carga como en la formulación clásica (Dantzig y Ramser, 1959). Se incorpora además una restricción propia del problema abordado: la obligación de que cada ruta se conecte con el corredor interurbano a través de un acceso a autopista, representando explícitamente la transición entre el recorrido local en Colina y el tramo hacia Santiago.

A diferencia de un problema de ruta mínima, que conecta un origen con un destino sin obligación de visitar puntos intermedios, este modelo exige visitar un conjunto

específico de paraderos antes de llegar al destino. Esta restricción tiene su origen en el problema del vendedor viajero (*Traveling Salesman Problem*, TSP), que busca determinar la secuencia en que un conjunto de puntos debe visitarse de manera que se minimice el tiempo o la distancia total del recorrido.

Las rutas generadas representan trazados base del sistema propuesto, pero no corresponden necesariamente a la cantidad final de buses requeridos. En la práctica, una misma ruta podría ser atendida por múltiples vehículos, dependiendo de la frecuencia requerida, la demanda observada y la disponibilidad de flota.

Los tiempos de viaje entre nodos se obtienen mediante OpenRouteService, por lo que el modelo no trabaja con distancias en línea recta, sino con tiempos de desplazamiento estimados sobre la red vial existente. Además, se incorpora un tratamiento previo de los puntos ubicados en pasajes sin salida, con el fin de evitar recorridos poco prácticos en los que el vehículo deba ingresar y salir por la misma vía.

Conjuntos

P : Conjunto de paraderos seleccionados por el modelo de localización, $i \in P$.

E : Conjunto de accesos a autopista disponibles, $i \in E$.

R : Conjunto máximo de rutas teóricas disponibles para construir la solución, $r \in R$.

$D = P \cup E$: Conjunto de nodos intermedios del modelo. Incluye los paraderos seleccionados y los accesos a autopista.

$\{0\}$: Nodo de origen de las rutas.

$\{F\}$: Nodo de destino de las rutas.

$N = D \cup \{0, F\}$: Conjunto total de nodos del modelo.

A : Conjunto de arcos posibles entre nodos, $(i, j) \in A$. Por construcción, no se permite volver al nodo de origen ni salir desde el nodo de destino. Además, el nodo de destino solo puede ser alcanzado desde un acceso a autopista.

$O(i) = \{j \in N : (i, j) \in A\}$: Conjunto de nodos a los que se puede ir directamente desde el nodo i , $\forall i \in N$.

$I(i) = \{j \in N : (j, i) \in A\}$: Conjunto de nodos desde los cuales se puede llegar directamente al nodo i , $\forall i \in N$.

Parámetros

t_{ij} : Tiempo de viaje entre el nodo i y el nodo j , obtenido sobre la red vial mediante OpenRouteService, $\forall (i, j) \in A$.

s_i : Tiempo de detención en el paradero i , $\forall i \in P$.

$T_{\text{máx}}$: Tiempo máximo permitido para cada ruta.

Variables de decisión

$$x_{ijr} = \begin{cases} 1, & \text{si la ruta } r \text{ utiliza el arco } (i, j) \\ 0, & \text{en otro caso} \end{cases} \quad \forall (i, j) \in A, r \in R$$

donde x_{ijr} indica si la ruta r viaja directamente desde el nodo i hacia el nodo j .

$$z_r = \begin{cases} 1, & \text{si la ruta } r \text{ es utilizada} \\ 0, & \text{en otro caso} \end{cases} \quad \forall r \in R$$

donde z_r indica si la ruta r forma parte de la solución final.

Función objetivo

El modelo busca minimizar el tiempo total de operación de las rutas generadas, considerando tanto el tiempo de viaje entre nodos como el tiempo de detención en los paraderos visitados:

$$\text{mín } Z = \sum_{r \in R} \left(\sum_{(i,j) \in A} t_{ij} x_{ijr} + \sum_{i \in P} s_i \sum_{j \in I(i)} x_{jir} \right)$$

El primer término corresponde al tiempo total de desplazamiento sobre la red vial, mientras que el segundo incorpora el tiempo de detención en cada paradero visitado. En este caso, el conjunto R representa una cantidad máxima de rutas disponibles, pero el modelo no está obligado a utilizarlas todas. La variable z_r permite identificar cuáles rutas se activan efectivamente dentro de la solución.

E

Restricciones

(1) Cobertura de todos los paraderos seleccionados:

$$\sum_{r \in R} \sum_{j \in I(i)} x_{jir} = 1 \quad \forall i \in P$$

Cada paradero seleccionado es visitado exactamente una vez por alguna de las rutas generadas.

(2) Salida desde el nodo de origen:

$$\sum_{j \in O(0)} x_{0jr} = z_r \quad \forall r \in R$$

Si una ruta r es utilizada, debe salir exactamente una vez desde el nodo de origen; si no es utilizada, no puede salir desde dicho nodo.

(3) Llegada al nodo de destino a través de un acceso a autopista:

$$\sum_{i \in I(F)} x_{iFr} = z_r \quad \forall r \in R$$

Esta restricción asegura que cada ruta utilizada finalice exactamente una vez en el nodo de destino. Dado que el destino solo puede ser alcanzado desde un acceso a autopista, esta condición obliga a que toda ruta activa incorpore la conexión entre el tramo local en Colina y el tramo hacia Santiago.

(4) Continuidad de las rutas:

$$\sum_{j \in I(i)} x_{jir} = \sum_{j \in O(i)} x_{ijr} \quad \forall i \in D, r \in R$$

Si una ruta entra a un nodo intermedio, también debe salir desde ese mismo nodo, asegurando la continuidad del recorrido.

(5) Visita mínima de paraderos por ruta utilizada:

$$\sum_{i \in P} \sum_{j \in I(i)} x_{jir} \geq z_r \quad \forall r \in R$$

Esta restricción impide que se active una ruta sin visitar al menos un paradero.

(6) Tiempo máximo por ruta:

$$\sum_{(i,j) \in A} t_{ij} x_{ijr} + \sum_{i \in P} s_i \sum_{j \in I(i)} x_{jir} \leq T_{\max} z_r \quad \forall r \in R$$

Esta restricción limita la duración de cada ruta utilizada, considerando tanto el tiempo de desplazamiento entre nodos como el tiempo de detención en los paraderos visitados. Si una ruta no es utilizada, es decir, si $z_r = 0$, entonces no puede incorporar arcos ni paraderos, por lo que su tiempo asociado queda forzado a cero.

(7) Orden de uso de rutas:

$$z_r \geq z_{r+1} \quad \forall r = 1, \dots, |R| - 1$$

Esta restricción de ruptura de simetría reduce soluciones equivalentes, forzando que las rutas se activen en orden.

(8) Eliminación de subtours:

$$\sum_{i \in S} \sum_{\substack{j \in S \\ (i,j) \in A}} x_{ijr} \leq |S| - 1 \quad \forall S \subseteq P, |S| \geq 2, \forall r \in R$$

Esta familia de restricciones evita la formación de ciclos internos entre paraderos dentro de una misma ruta. Dado que el número de subconjuntos $S \subseteq P$ crece exponencial-

mente, estas restricciones se incorporan de forma dinámica mediante un procedimiento de planos cortantes. Primero se resuelve el modelo sin todas las restricciones de subtour, luego se identifican ciclos en la solución obtenida, se agregan los cortes correspondientes y se repite el proceso hasta que no se detecten subtours.

(9) Dominios de las variables:

$$x_{ijr} \in \{0, 1\} \quad \forall (i, j) \in A, r \in R$$

$$z_r \in \{0, 1\} \quad \forall r \in R$$

4.2.1. Preparación de los insumos del modelo de ruteo

La implementación del modelo de ruteo requiere consolidar previamente los elementos espaciales que serán utilizados para construir las rutas alimentadoras. Estos insumos corresponden a los paraderos seleccionados por el modelo de localización, los accesos a autopista, el nodo de origen y el nodo de destino. Cada elemento se trabaja mediante coordenadas geográficas, lo que permite integrarlo con herramientas de cálculo de rutas y visualizar posteriormente los resultados en sistemas de información geográfica.

En esta etapa se utilizó como base el escenario seleccionado del modelo de localización, correspondiente a $p = 60$, obteniéndose un total de 60 paraderos, 2 accesos a autopista, 1 nodo de origen y 1 nodo de destino. Previo a la ejecución del modelo se realizó además una revisión espacial de los puntos, descrita en la sección siguiente.

Tabla 2: Resumen de insumos espaciales del modelo de ruteo

Elemento	Símbolo	Valor utilizado
Paraderos utilizados en el ruteo	P	60
Accesos a autopista	E	2
Nodo de origen	$\{0\}$	1
Nodo de destino	$\{F\}$	1
Nodos efectivos del modelo	N	64
Rutas máximas disponibles	R	6

Como se observa en la Tabla 2, el conjunto de rutas R corresponde a una cota superior definida para el modelo, no a una cantidad obligatoria de recorridos. La variable de activación de ruta permite que el modelo utilice solo aquellas rutas necesarias para cumplir las condiciones de cobertura, continuidad y tiempo máximo.

El nodo de destino fue definido en la estación Los Libertadores, debido a que corresponde a la estación de Metro más cercana a la comuna de Colina y permite conectar el sistema propuesto con el resto de la red de transporte metropolitano de Santiago. Por su parte, los accesos considerados cumplen el rol de vincular el recorrido local dentro de Colina con el tramo interurbano hacia Santiago mediante la Autopista Los Libertadores.

4.2.2. Construcción de la matriz de tiempos y arcos factibles

A partir de los nodos definidos se construyó una matriz de tiempos de viaje entre pares de puntos mediante OpenRouteService. A diferencia de una matriz basada en distancia euclidiana, esta matriz considera desplazamientos sobre la red vial existente, incluyendo calles locales, vías principales, caletas y accesos hacia autopista.

El uso de tiempos sobre red vial permite representar de mejor manera las condiciones de desplazamiento de un vehículo. Cada elemento de la matriz corresponde al tiempo estimado para viajar desde un nodo i hacia un nodo j , y se expresa en minutos. De forma complementaria, también se almacena la distancia asociada a cada tramo, medida en kilómetros.

Tabla 3: Parámetros principales utilizados en el modelo de ruteo		
Parámetro	Símbolo	Valor utilizado
Perfil de ruteo	–	Vehículo
Tiempo máximo por ruta	$T_{\text{máx}}$	70 min
Tiempo de detención por paradero	s_i	0,5 min
Registros válidos de tiempo y distancia	t_{ij}	2.602
Unidad de tiempo	–	Minutos
Unidad de distancia	–	Kilómetros
Radio de ajuste a red vial	–	350 m

La Tabla 3 resume los principales parámetros utilizados en la construcción de la

instancia de ruteo. El tiempo máximo por ruta fue fijado en 70 minutos, con el objetivo de evitar recorridos excesivamente largos. Además, se incorporó un tiempo de detención de 0,5 minutos por paradero, asociado al ascenso de pasajeros en cada punto de recolección.

Junto con el cálculo de tiempos, se definió el conjunto de arcos factibles del modelo. Esta etapa es necesaria porque no todas las conexiones entre nodos son admisibles desde el punto de vista operacional. En particular, no se permite retornar al nodo de origen, salir desde el nodo de destino ni conectar directamente cualquier paradero con el destino final. La llegada al destino se restringe a los accesos de autopista, representando así la transición entre el recorrido alimentador local y el corredor interurbano hacia Santiago.

4.2.3. Tratamiento de pasajes sin salida y coherencia espacial

Uno de los ajustes incorporados al procedimiento corresponde al tratamiento de puntos ubicados en pasajes sin salida. Estos puntos pueden generar soluciones matemáticamente factibles, pero poco convenientes en la práctica, debido a que obligan al vehículo a realizar maniobras de entrada y salida por una misma calle.

Para identificar estos casos se utiliza un criterio combinado: por una parte, un diagnóstico basado en la matriz de tiempos calculada mediante OpenRouteService, que evalúa la similitud entre los tiempos de entrada y salida respecto del vecino más cercano, junto con el grado de aislamiento del punto frente a otras alternativas de conexión; por otra parte, la confirmación visual del trazado resultante sobre la cartografía, dado que un criterio puramente estadístico puede no capturar todos los casos relevantes ni descartar falsos positivos. La decisión final de qué paraderos corregir se basa en esta revisión visual, utilizando el diagnóstico estadístico como herramienta de apoyo para acotar los puntos a inspeccionar.

Tabla 4: Criterios utilizados para el diagnóstico de pasajes sin salida

Criterio	Valor utilizado
Umbral de similitud entrada-salida	0,75
Umbral de aislamiento respecto del segundo vecino	1,50
Tiempo máximo al vecino cercano	5,0 min

A diferencia de una restricción adicional dentro del modelo, este tratamiento corrige la representación espacial del dato de entrada, sin modificar la formulación matemática del modelo de ruteo. Dado que el modelo se resuelve de manera exacta para cada instancia, la eliminación de uno o más paraderos no equivale a remover dichos puntos de la solución previamente obtenida, sino que da lugar a una nueva resolución completa del problema sobre el conjunto de paraderos restante. En consecuencia, la nueva solución puede diferir de la anterior tanto en la secuencia de visita como en la distribución de los paraderos entre las rutas optimizadas.

4.2.4. Procedimiento de resolución computacional

El modelo fue implementado en Python mediante Pyomo. En primer lugar, se cargaron los nodos georreferenciados y se clasificaron según su función dentro del modelo: paraderos, accesos, origen y destino. Luego se construyó la matriz de tiempos y distancias entre nodos, utilizando OpenRouteService como herramienta de cálculo sobre red vial.

Posteriormente, se definieron los conjuntos, parámetros y variables del modelo, incorporando las restricciones de cobertura, continuidad, activación de rutas, conexión con accesos a autopista, tiempo máximo por ruta y eliminación de subtours. Estas últimas se agregaron de manera iterativa mediante planos cortantes, incorporando únicamente los cortes necesarios cuando se detectaron ciclos internos en la solución.

Tabla 5: Parámetros computacionales de resolución

Elemento	Valor utilizado
Lenguaje de implementación	Python
Herramienta de optimización	Pyomo
Servicio de matriz vial	OpenRouteService
Función objetivo utilizada	Minimización del tiempo total
Límite de tiempo por intento	45 s
Brecha de optimalidad	2 %
Máximo de iteraciones de cortes	25

La Tabla 5 resume las principales condiciones computacionales utilizadas en la resolución. La función objetivo corresponde a la minimización del tiempo total de opera-

ción, por lo que el modelo no minimiza explícitamente la cantidad de rutas. En cambio, el conjunto de rutas disponibles opera como una cota superior y el modelo activa aquellas que resultan necesarias para satisfacer las restricciones del problema.

4.2.5. Evaluación de las rutas generadas

Los resultados del modelo entregan una propuesta preliminar de rutas alimentadoras para conectar los puntos de recolección seleccionados con los corredores hacia Santiago. Para cada ruta se obtiene la secuencia de paraderos visitados, el acceso utilizado, la distancia total, el tiempo de viaje, el tiempo de detención y el tiempo total de operación.

La evaluación de las rutas se realiza considerando tanto indicadores cuantitativos como criterios espaciales. Entre los principales elementos revisados se encuentran la cantidad de rutas activadas, el número de paraderos asignados a cada recorrido, el tiempo total de operación y la coherencia territorial del trazado.

También se revisa que las rutas no incorporen desvíos innecesarios, entradas a pasajes sin salida o secuencias poco razonables desde el punto de vista vial. Esta revisión es necesaria porque el modelo trabaja con una matriz de tiempos entre puntos de interés, por lo que no reemplaza completamente una validación operacional en terreno.

Finalmente, las rutas obtenidas deben interpretarse como una base inicial para el diseño del sistema. La propuesta permite transformar los puntos óptimos de recolección en recorridos preliminares, cuya operación definitiva requeriría analizar posteriormente frecuencias, flota, horarios, capacidad de vehículos y validación práctica de los trazados.

5. Resultados y discusión

5.1. Resultados de la matriz de distancias

La matriz de distancias constituye uno de los insumos principales del modelo de localización, ya que permite relacionar cada nodo de demanda con cada punto candidato

definido en el área de estudio. En este caso, se trabajó con 991 nodos de demanda y 70 puntos candidatos, por lo que la matriz d_{ij} quedó compuesta por 991×70 combinaciones posibles. Cada elemento d_{ij} representa la distancia, medida en kilómetros, entre el nodo de demanda i y el punto candidato j .

Las distancias fueron calculadas considerando un perfil de desplazamiento peatonal, debido a que el objetivo del modelo es evaluar la accesibilidad de los usuarios hacia los puntos de recolección. De esta forma, la matriz no representa una distancia en línea recta, sino una distancia asociada a la red caminable disponible. Esto permite aproximar de mejor manera el esfuerzo real que debe realizar una persona para llegar desde su zona de origen hasta un punto de acceso al sistema.

Tabla 6: Resumen de la matriz de distancias

Elemento	Resultado
Nodos de demanda	991
Puntos candidatos	70
Dimensión de la matriz	991×70
Perfil de ruta utilizado	Caminata
Unidad de distancia	Kilómetros
Unidad de tiempo	Minutos
Radio óptimo de cobertura	$R_1 = 0,4 \text{ km}$
Radio deseable de cobertura	$R_2 = 0,5 \text{ km}$
Nodos sin cobertura en R_1	193
Nodos sin cobertura en R_2	97
Cobertura geográfica máxima en R_1	79,48 %
Cobertura geográfica máxima en R_2	89,18 %

Como se observa en la Tabla 6, existen nodos de demanda que no poseen ningún punto candidato dentro de los radios de cobertura definidos. Esto implica que la cobertura máxima del modelo no depende únicamente de la cantidad de puntos seleccionados, sino también de la ubicación inicial de los puntos candidatos. En particular, la cobertura geográfica máxima alcanzable corresponde a 79,48 % para R_1 y 89,18 % para R_2 , por lo que una cobertura del 100 % no resulta factible con la configuración actual.

Este resultado es relevante porque permite interpretar correctamente los resultados posteriores del modelo. Si ciertos sectores de demanda se encuentran alejados de todos los

puntos candidatos, estos no podrán ser cubiertos aunque se aumente el número de puntos seleccionados. Por lo tanto, la matriz de distancias no solo permite alimentar el modelo de optimización, sino también identificar una limitación territorial previa asociada al diseño del conjunto de alternativas disponibles.

5.2. Resultados del modelo de localización

El modelo de localización fue evaluado considerando distintos escenarios según la cantidad de puntos de recolección seleccionados. En particular, se analizaron los casos $p = 20$, $p = 30$, $p = 40$, $p = 50$, $p = 60$ y $p = 70$. Para cada escenario, se obtuvo el valor de la función objetivo, la cobertura alcanzada en los radios R_1 y R_2 , la cantidad de nodos cubiertos y la utilización promedio de los puntos habilitados.

Tabla 7: Resumen de resultados por escenario

p	Objetivo	Cob. R_1	Cob. R_2	Nodos R_1	Nodos R_2	Util. prom.
20	6082,77	61,94 %	79,16 %	613	781	13,17 %
30	6886,48	71,78 %	86,31 %	721	863	9,58 %
40	7243,73	76,64 %	88,53 %	769	887	7,37 %
50	7405,22	79,02 %	89,15 %	791	893	5,93 %
60	7433,61	79,48 %	89,18 %	798	894	4,95 %
70	7433,61	79,48 %	89,18 %	798	894	4,24 %

De acuerdo con la Tabla 7, el mejor desempeño general se obtiene en el escenario $p = 60$, ya que alcanza el mayor valor de la función objetivo, igual a $Z = 7433,61$. Este escenario cubre 798 nodos dentro del radio óptimo R_1 y 894 nodos dentro del radio deseable R_2 , lo que equivale a coberturas de 79,48 % y 89,18 %, respectivamente. El escenario $p = 70$ obtiene exactamente el mismo resultado, por lo que $p = 60$ se prefiere por representar la misma cobertura con menos puntos habilitados.

El comportamiento de los resultados muestra que la cobertura crece de forma monótona a medida que aumenta p , sin retrocesos: cada escenario cubre una cantidad de nodos igual o mayor al anterior. Este crecimiento, sin embargo, presenta rendimientos decrecientes marcados: entre $p = 20$ y $p = 50$ la cobertura en R_2 aumenta en un 9,99 %,

mientras que entre $p = 50$ y $p = 60$ el incremento es de apenas 0,03 %. A partir de $p = 60$ la cobertura deja de mejorar por completo, ya que $p = 60$ y $p = 70$ obtienen resultados idénticos. Esto ocurre porque $p = 60$ alcanza exactamente la cobertura geográfica máxima identificada en la matriz de distancias (79,48 % en R_1 y 89,18 % en R_2): no existen más nodos que puedan incorporarse a la cobertura, independientemente de cuántos puntos candidatos adicionales se habiliten, ya que 193 nodos de demanda no cuentan con ningún candidato dentro de R_1 y 97 nodos no cuentan con ningún candidato dentro de R_2 .

Respecto de la utilización promedio de los puntos, esta disminuye de forma monótona a medida que aumenta p , pasando de 13,17 % en $p = 20$ a 4,24 % en $p = 70$. Esto es consistente con que la capacidad de cada punto candidato se definió como un valor fijo, independiente del número de puntos habilitados: al aumentar p , la misma demanda total se reparte entre más puntos, reduciendo la utilización promedio de cada uno sin que esto refleje una limitación real del sistema.

A partir de lo anterior, el escenario seleccionado corresponde a $p = 60$, ya que alcanza el mayor valor de la función objetivo y coincide exactamente con el límite geográfico máximo alcanzable dado el conjunto actual de puntos candidatos. Por lo tanto, las mejoras adicionales de cobertura no pueden lograrse aumentando la cantidad de puntos seleccionados, sino que requieren incorporar nuevos puntos candidatos en los sectores que actualmente no cuentan con ninguna alternativa dentro de los radios de cobertura definidos.

5.3. Comparación espacial de cobertura

La comparación espacial permite observar cómo cambia la distribución de los puntos de acceso entre la situación actual y la solución propuesta por el modelo. Este análisis es relevante porque no basta con conocer el porcentaje de cobertura obtenido, sino que también es necesario identificar dónde se ubican los puntos y qué sectores quedan mejor atendidos.

En la Imagen 4 se presenta la distribución de los paraderos actuales considerados en el área de estudio. Se observa que estos se concentran principalmente en torno a la Caletera Oriente General San Martín y a los ejes viales principales. Esta configuración responde a la infraestructura existente, pero no necesariamente a la distribución espacial de la demanda. Por esta razón, algunos sectores residenciales pueden quedar alejados de los puntos de acceso actuales, generando mayores distancias de caminata para los usuarios.

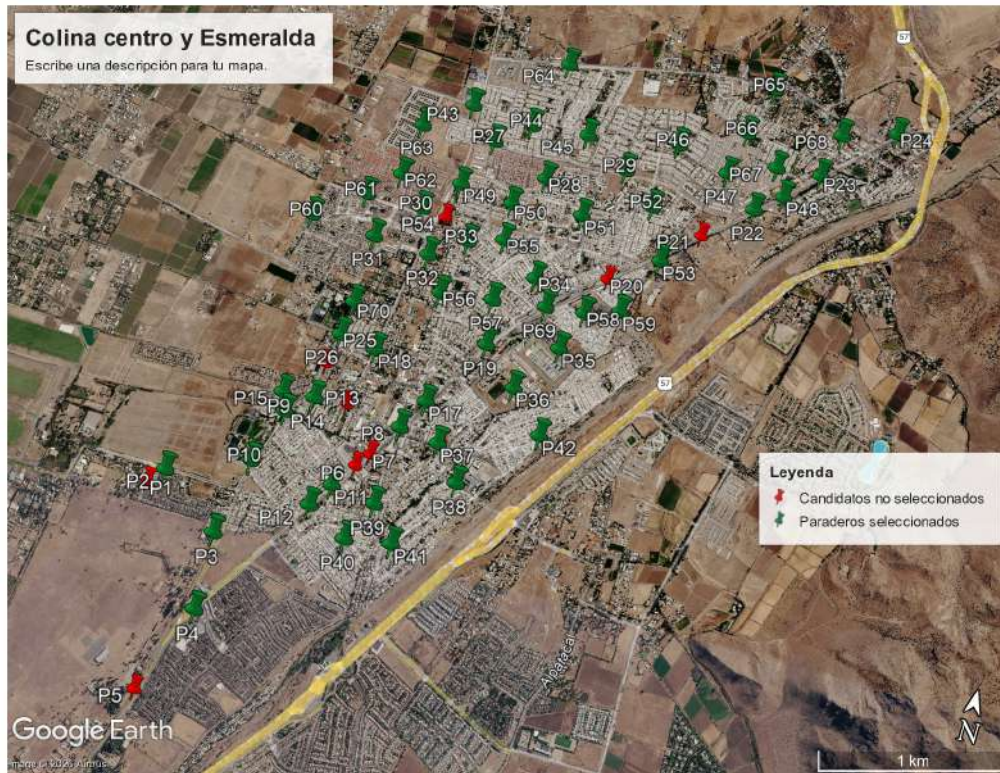


Imagen 4: Distribución espacial de los paraderos actuales

La situación anterior permite evidenciar una de las principales limitaciones del sistema actual: los puntos de acceso se encuentran asociados principalmente a la red vial existente, pero no necesariamente se distribuyen de forma equilibrada respecto de los nodos de demanda. En consecuencia, la cobertura puede ser adecuada para sectores cercanos a los corredores principales, pero menos favorable para zonas interiores o residenciales.

En la Imagen 5 se presenta la solución obtenida mediante el modelo de localización para el escenario seleccionado ($p = 60$). A diferencia de la situación actual, los puntos propuestos se definen considerando la cercanía con los nodos de demanda, los radios

de cobertura y la capacidad de atención de cada punto. De esta manera, el modelo de localización busca mejorar la accesibilidad peatonal y aumentar la cantidad de usuarios cubiertos dentro de las distancias definidas.



Imágen 5: Distribución espacial de los puntos seleccionados en el escenario propuesto ($p = 60$)

Al comparar ambas imágenes, se aprecia que la propuesta del modelo no busca replicar la ubicación de los paraderos actuales, sino generar una distribución más coherente con la localización de la demanda. Esto permite mejorar la relación entre los puntos de acceso y las zonas residenciales analizadas, especialmente en aquellos sectores donde existe una mayor concentración de nodos.

No obstante, la solución propuesta también presenta limitaciones. Un total de 193 nodos de demanda no cuenta con ningún punto candidato dentro del radio óptimo R_1 , y 97 nodos no cuenta con ningún candidato dentro del radio deseable R_2 , debido a que no existen puntos candidatos cercanos disponibles para ser seleccionados en esos sectores. Por lo tanto, estas zonas no pueden ser cubiertas únicamente mediante la optimización

del modelo actual, sin importar cuántos puntos adicionales se habiliten. Para mejorar la cobertura en esos sectores, sería necesario incorporar nuevos puntos candidatos y volver a ejecutar el modelo.

En síntesis, la comparación espacial muestra que la solución propuesta mejora la distribución de los puntos de acceso respecto de la situación actual, ya que considera explícitamente la ubicación de la demanda y las distancias peatonales. Sin embargo, también evidencia que el potencial de cobertura está condicionado por el conjunto inicial de puntos candidatos, por lo que futuras mejoras deben enfocarse en ampliar y ajustar dichas alternativas de localización.

5.4. Resultados del modelo de ruteo

A partir del escenario seleccionado en el modelo de localización, correspondiente a $p = 60$, se construyó una propuesta preliminar de rutas alimentadoras orientadas a conectar los puntos de recolección con un corredor principal hacia Santiago. Para esta etapa, el nodo de destino se definió en la estación intermodal Los Libertadores, por corresponder a la estación de Metro más cercana a la comuna de Colina y por facilitar la conexión con el resto de la red de transporte metropolitano.

El procesamiento previo al modelo de ruteo consideró 64 puntos georreferenciados, compuestos por 60 paraderos seleccionados, 2 accesos a autopista, 1 nodo de origen y 1 nodo de destino. A partir de ellos se calcularon 3.722 registros válidos de tiempo y distancia mediante OpenRouteService, utilizados posteriormente como insumo para el modelo.

Tabla 8: Resumen del procesamiento previo al modelo de ruteo

Elemento	Resultado
Paraderos seleccionados en localización	60
Accesos considerados	2
Nodo de origen	1
Nodo de destino (Los Libertadores)	1
Registros válidos de tiempo y distancia	3.722

La solución final quedó compuesta por tres rutas. La Ruta 3 corresponde al trazado actualmente operativo en la comuna, incorporado como ruta fija, es decir, no sujeta a optimización, dado que su estructura ya se encuentra validada operacionalmente. Esta ruta fija está conformada por 7 paraderos; esta cantidad puede considerarse poco realista, pero normalmente los buses de transporte realizan detenciones a lo largo de la ruta sin necesidad de tener un paradero, lo cual es poco eficiente y riesgoso para todos los involucrados. Los 53 paraderos restantes fueron los efectivamente optimizados por el modelo y distribuidos en las Rutas 1 y 2 como recorridos alimentadores complementarios.

Tabla 9: Resumen de rutas consideradas en la propuesta

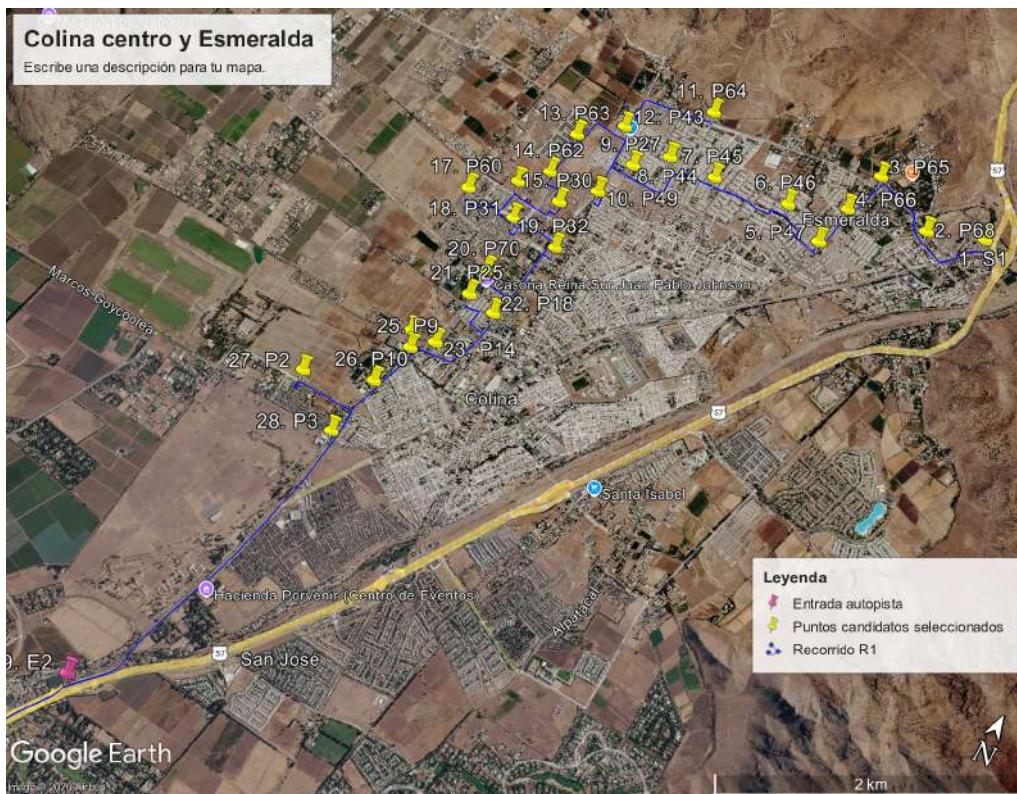
Ruta	Tipo	Paraderos	Acceso	Distancia km	Viaje min	Detención min	Total min
Ruta 1	Optimizada	27	E2	32,95	54,16	13,50	67,66
Ruta 2	Optimizada	26	E1	34,83	56,30	13,00	69,30
Ruta 3	Fija actual	7	E2	24,11	25,15	3,50	28,65

La Tabla 9 muestra que las rutas optimizadas concentran la mayor parte de los paraderos considerados en la etapa de ruteo. La Ruta 1 incorpora 27 paraderos y alcanza un tiempo total de 67,66 minutos, mientras que la Ruta 2 cubre 26 paraderos y presenta un tiempo total de 69,30 minutos. Ambas rutas se encuentran muy cercanas al tiempo máximo permitido de 70 minutos, lo que indica que la restricción de duración actúa como un mecanismo indirecto de consolidación: aunque la función objetivo no minimiza explícitamente la cantidad de rutas, el límite temporal impide concentrar todos los paraderos optimizados en un único recorrido sin superar el umbral operacional definido.

La Ruta 3, en cambio, corresponde al trazado actual mantenido como ruta fija dentro de la propuesta. Su inclusión no debe interpretarse como una decisión del modelo de optimización, sino como una decisión metodológica previa orientada a conservar un eje operativo existente y complementar su funcionamiento mediante nuevas rutas alimentadoras. De esta forma, la propuesta combina conocimiento operacional previo con optimización matemática aplicada a los paraderos restantes.

Adicionalmente, debe considerarse que los tiempos utilizados en el modelo no incorporan condiciones dinámicas de tráfico vial. La matriz de tiempos fue construida

a partir de estimaciones sobre la red vial, pero no considera congestión en horas punta, incidentes, variaciones por día de la semana ni demoras asociadas a la operación real en intersecciones. Por lo tanto, los tiempos obtenidos deben interpretarse como una aproximación base para comparar y construir rutas, y no como tiempos definitivos de operación. En la práctica, los tiempos de traslado podrían aumentar de manera significativa, especialmente en los tramos de conexión hacia la Autopista Los Libertadores y en el acceso a la estación Los Libertadores.

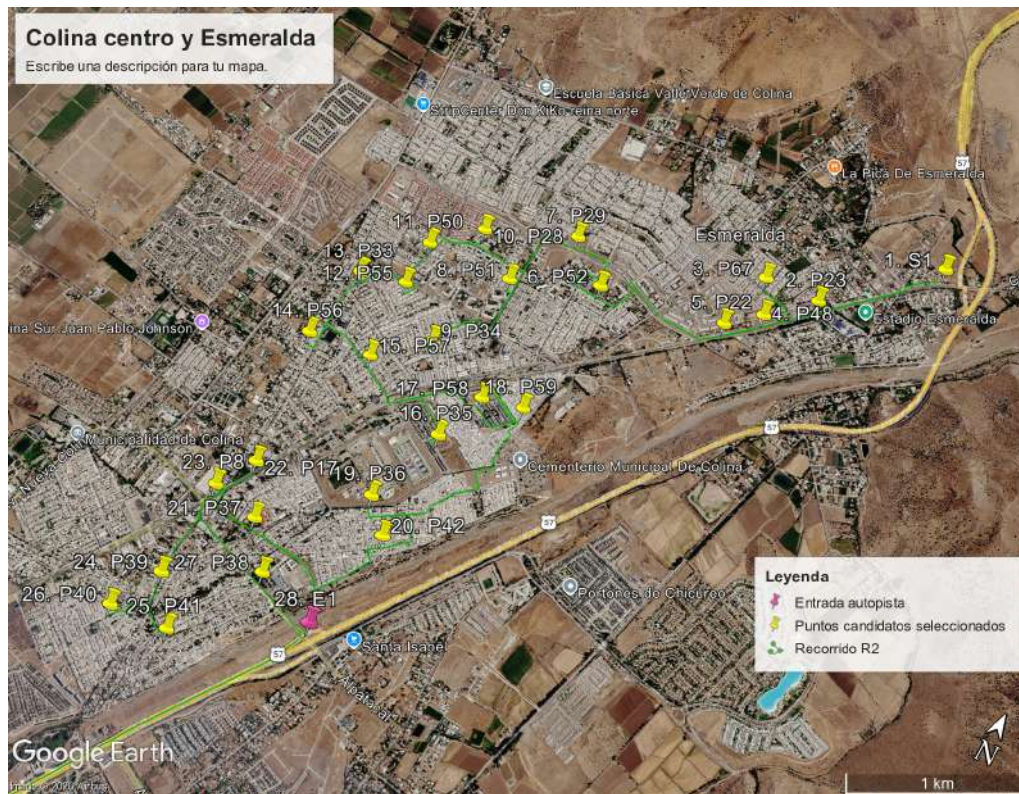


Imágen 6: Trazado generado para la Ruta 1

La Ruta 1, presentada en la Imágen 6, recorre principalmente sectores residenciales del norte, centro y poniente del área urbana de Colina. Su trazado agrupa la mayor cantidad de paraderos optimizados, lo que explica que presente el mayor tiempo de detención acumulado. Esta ruta utiliza el acceso E2 para incorporarse posteriormente al corredor interurbano.

Desde el punto de vista funcional, la Ruta 1 cumple un rol alimentador importante, ya que permite recoger demanda desde sectores interiores que no se encuentran directa-

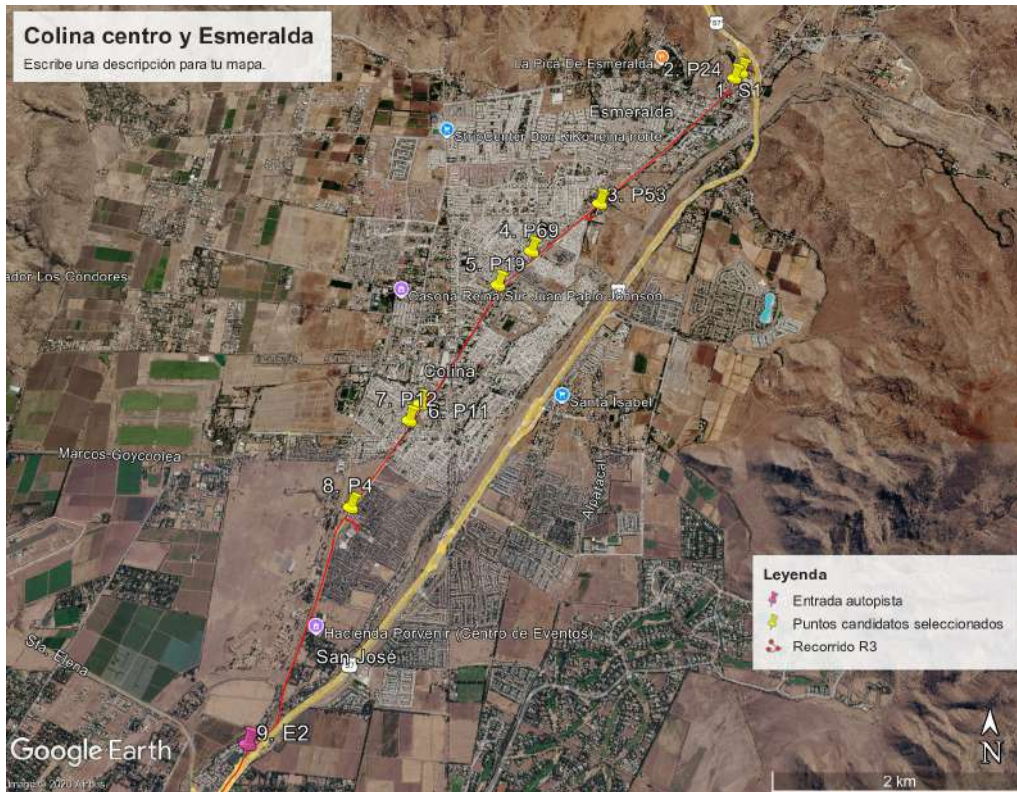
mente sobre el eje estructurante principal. En consecuencia, su valor no radica en replicar el recorrido actual, sino en complementar la red propuesta mediante la incorporación de zonas residenciales con menor accesibilidad directa.



Imágen 7: Trazado generado para la Ruta 2

La Ruta 2, mostrada en la Imágen 7, cubre 26 paraderos y utiliza el acceso E1. Su recorrido se desarrolla principalmente en el sector nororiente y centro de Colina, conectando luego con la red principal a través de un acceso distinto al de la Ruta 1. Su distancia total es la mayor de las tres rutas, y su tiempo total también resulta el más alto, muy cercano al límite de 70 minutos.

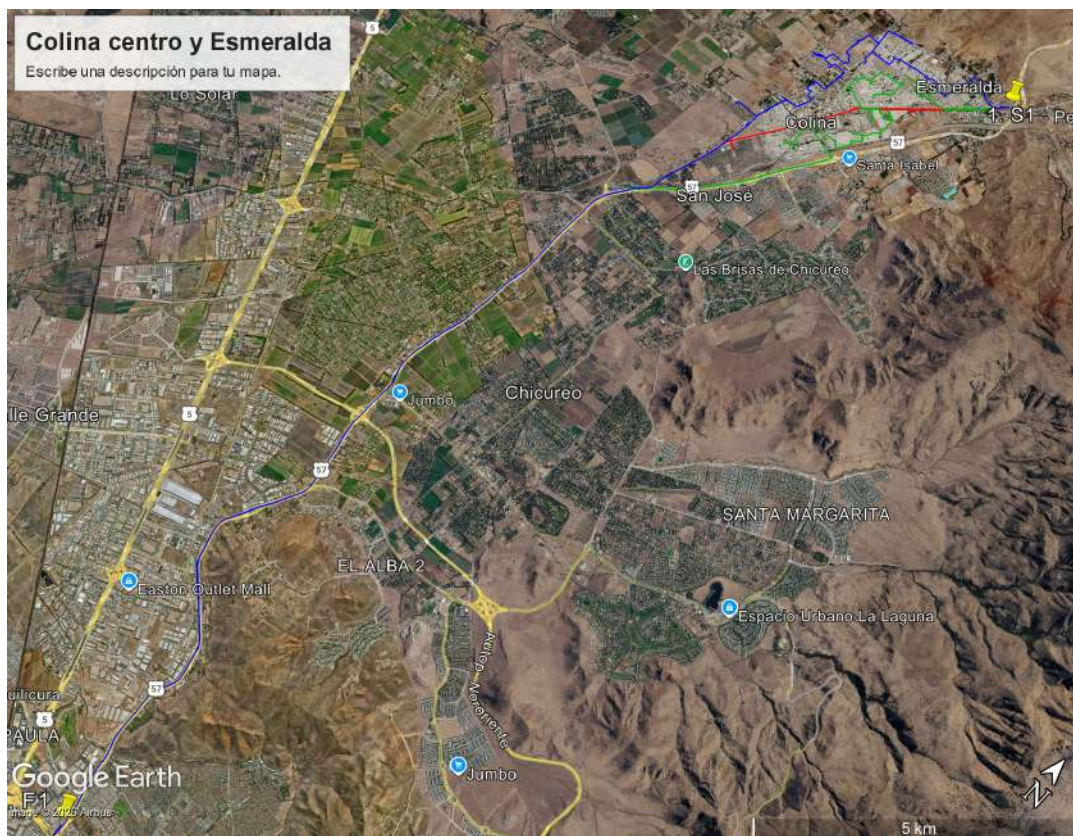
Este resultado muestra que la duración de una ruta no depende solo de la distancia recorrida, sino también de la estructura vial del sector, del número de paraderos servidos y del acceso utilizado para conectar con la Autopista Los Libertadores. En este caso, la Ruta 2 permite diversificar territorialmente la salida desde Colina hacia Santiago.



Imágen 8: Trazado correspondiente a la Ruta 3, incorporada como ruta fija actual

La Ruta 3, presentada en la Imágen 8, corresponde al recorrido actualmente operativo considerado como ruta fija dentro de la propuesta. Está compuesta por 7 paraderos y un tiempo total estimado de 28,65 minutos. Su estructura es más lineal que la de las rutas optimizadas y se vincula con el acceso E2, manteniendo una lógica territorial coherente con el funcionamiento actual del servicio.

La incorporación de esta ruta fija permite preservar un trazado existente y concentrar la optimización en los sectores no cubiertos por dicho eje. En este sentido, las Rutas 1 y 2 no reemplazan la operación actual, sino que la complementan mediante recorridos alimentadores orientados a ampliar la cobertura territorial del sistema.



Imágen 9: Esquema conjunto de las rutas propuestas y su conexión con Santiago mediante la Autopista Los Libertadores hasta la estación Los Libertadores

La Imágen 9 permite visualizar de forma integrada la estructura general del sistema propuesto. En ella se observa cómo las rutas ubicadas dentro de Colina convergen hacia los accesos definidos y se articulan posteriormente con la Autopista Los Libertadores para conectar con la estación Los Libertadores. Esta representación permite evidenciar que la propuesta no se limita a organizar puntos de recolección dentro de la comuna, sino que los integra con una conexión interurbana concreta hacia Santiago.

5.5. Análisis operacional de las rutas propuestas

Los resultados obtenidos muestran que la propuesta de ruteo transforma los puntos seleccionados por el modelo de localización en una estructura preliminar compuesta por tres recorridos diferenciados. La Ruta 1 y la Ruta 2 corresponden a recorridos optimizados, diseñados para complementar la cobertura territorial del sistema, mientras que la Ruta 3

corresponde al trazado actual incorporado como ruta fija.

Desde el punto de vista temporal, la Ruta 1 y la Ruta 2 presentan duraciones cercanas al límite máximo permitido, en tanto que la Ruta 3 presenta una duración menor a 30 minutos. Esta diferencia se explica tanto por la extensión territorial de cada recorrido como por la cantidad de paraderos incorporados. En particular, la Ruta 1 concentra el 45 % de los paraderos considerados en el ruteo, la Ruta 2 el 43,3 % y la Ruta 3 el 11,7 %.

El tiempo de detención también tiene un efecto relevante sobre el desempeño de las rutas. En la Ruta 1, las detenciones representan 13,5 minutos de los 67,66 minutos totales; en la Ruta 2 alcanzan 13,0 minutos de 69,30 minutos; y en la Ruta 3 llegan a 3,5 minutos de 28,65 minutos. Esto muestra que la cantidad de paradas tiene un impacto significativo sobre la duración final del recorrido, por lo que en una etapa posterior la estimación de frecuencias y flota deberá considerar no solo la longitud del trazado, sino también la carga operacional asociada al número de detenciones.

Espacialmente, la solución presenta una lógica coherente con la estructura urbana de Colina. Las rutas optimizadas recogen demanda desde barrios residenciales que hoy poseen menor proximidad a un eje troncal, mientras que la conexión final con la estación Los Libertadores se realiza a través de la Autopista Los Libertadores, aprovechando una infraestructura vial ya consolidada y una estación terminal que permite distribuir los viajes hacia distintos puntos de Santiago mediante la red de Metro.

La cercanía de la Ruta 1 y la Ruta 2 al límite de 70 minutos también permite interpretar por qué el modelo no consolidó todos los paraderos optimizados en un único recorrido. Si bien la función objetivo minimiza el tiempo total de operación y no penaliza directamente la apertura de rutas adicionales, el tiempo máximo por ruta limita la cantidad de puntos que pueden incorporarse en un mismo trazado. Por ello, la generación de dos rutas optimizadas resulta coherente con la restricción operacional definida.

5.6. Segunda iteración: corrección de paraderos en pasajes sin salida

Con el fin de mejorar la coherencia operacional de los trazados obtenidos, se realizó una segunda iteración del modelo de ruteo. En esta etapa se identificaron visualmente los paraderos cuyo trazado evidenciaba un patrón de entrada y salida por la misma vía, corrigiendo su ubicación hacia el tramo de la ruta principal más cercano cuando esto era posible, y excluyendo aquellos casos en que el punto se encontraba en el interior de un pasaje sin ninguna vía alternativa de salida. En total, se corrigió la ubicación de 22 paraderos y se excluyeron 8, quedando una instancia de 52 paraderos disponibles para el ruteo.

Tabla 10: Resumen del procesamiento previo a la segunda iteración

Elemento	Resultado
Paraderos disponibles para el ruteo	52
Accesos considerados	2
Nodo de origen	1
Nodo de destino (Los Libertadores)	1
Registros válidos de tiempo y distancia	2.810

La solución de esta segunda iteración se mantuvo compuesta por tres rutas, con la misma composición de paraderos en la ruta fija que en la iteración anterior. Los 45 paraderos restantes fueron optimizados y distribuidos entre las Rutas 1 y 2.

Tabla 11: Resumen de rutas de la segunda iteración

Ruta	Tipo	Paraderos	Acceso	Distancia km	Viaje min	Detención min	Total min
Ruta 1	Optimizada	25	E2	27,48	41,00	12,50	53,50
Ruta 2	Optimizada	20	E1	28,76	38,96	10,00	48,96
Ruta 3	Fija actual	7	E2	23,50	22,90	3,50	26,40

Tabla 12: Comparación entre la primera y la segunda iteración

Ruta	Δ paraderos	Δ tiempo total (min)	Reducción
Ruta 1	-2	-14,16	-20,9 %
Ruta 2	-6	-20,34	-29,4 %
Ruta 3 (fija)	0	-2,25	-7,9 %

Como se observa en la Tabla 12, la corrección de paraderos redujo el tiempo total de las tres rutas. La reducción resulta desproporcionada respecto de la cantidad de

paraderos removidos: en la Ruta 2, la eliminación del 23 % de los paraderos (6 de 26) produjo una reducción del 29,4 % en el tiempo total, por sobre lo que correspondería a una reducción proporcional. Esto es consistente con que los paraderos corregidos generaban un sobre costo de tiempo asociado al ingreso y salida por una misma vía, en lugar de aportar un tiempo de desplazamiento equivalente al resto de los paraderos de la ruta.

La Ruta 3, correspondiente a la ruta fija, también presentó una reducción de tiempo pese a mantener los mismos 7 paraderos, dado que dos de sus puntos fueron reubicados durante la corrección espacial, sin alterar la secuencia de visita definida para este recorrido.

Con los tiempos obtenidos en esta segunda iteración, la Ruta 1 y la Ruta 2 dejaron de encontrarse cercanas al límite máximo de 70 minutos por ruta. Sin embargo, la suma de los tiempos de ambas rutas optimizadas asciende a 102,46 minutos, valor que continúa superando ampliamente el límite permitido para una única ruta. Esto indica que la generación de dos rutas optimizadas no depende de que cada una se encuentre cerca del límite individual, sino de que el conjunto de paraderos optimizados no puede ser atendido dentro de una sola ruta sin superar la restricción de tiempo máximo definida en el modelo.

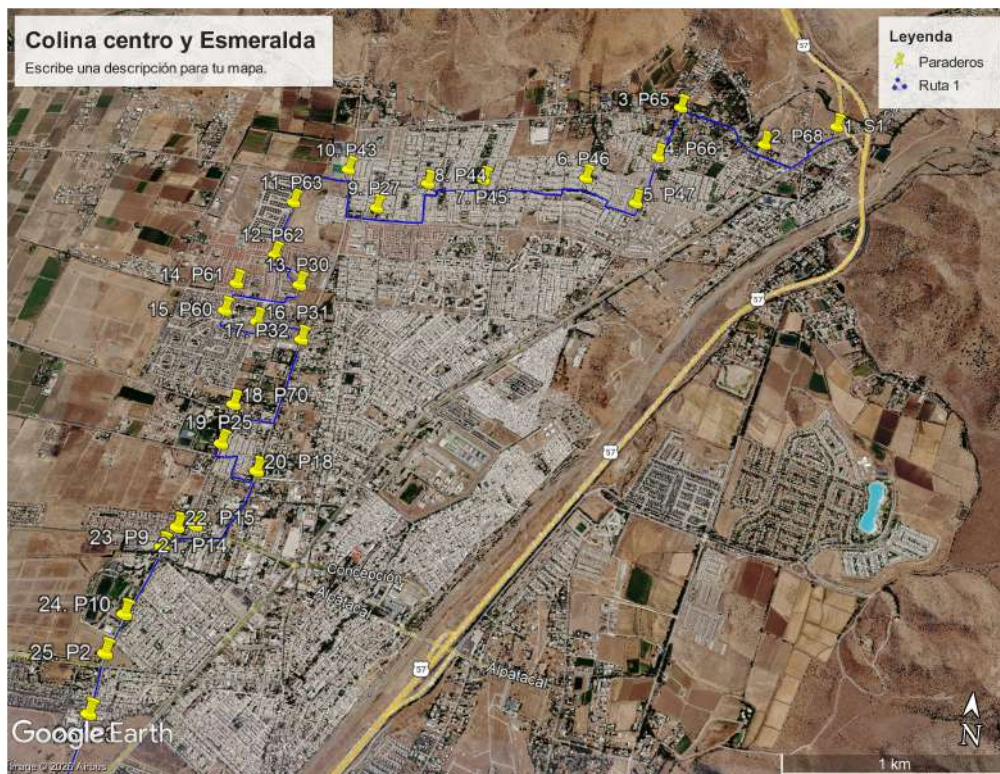
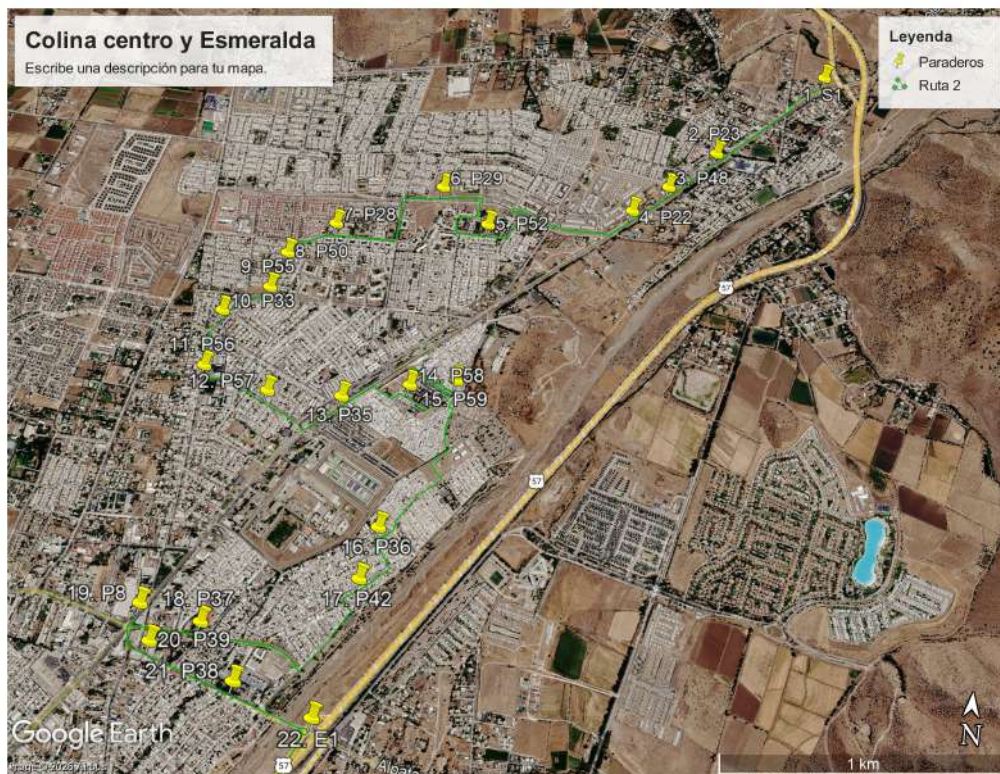


Imagen 10: Trazado generado para la Ruta 1, segunda iteración

La Ruta 1, presentada en la Imagen 10, mantiene su recorrido por los sectores residenciales del norte, centro y poniente del área urbana de Colina, con 25 paraderos tras la corrección espacial. Su estructura general es similar a la obtenida en la primera iteración, pero sin los tramos de entrada y salida asociados a los dos paraderos corregidos en esta ruta, lo que se refleja en una reducción de 14,16 minutos respecto del trazado anterior. Esta ruta continúa utilizando el acceso E2 para incorporarse al corredor interurbano.



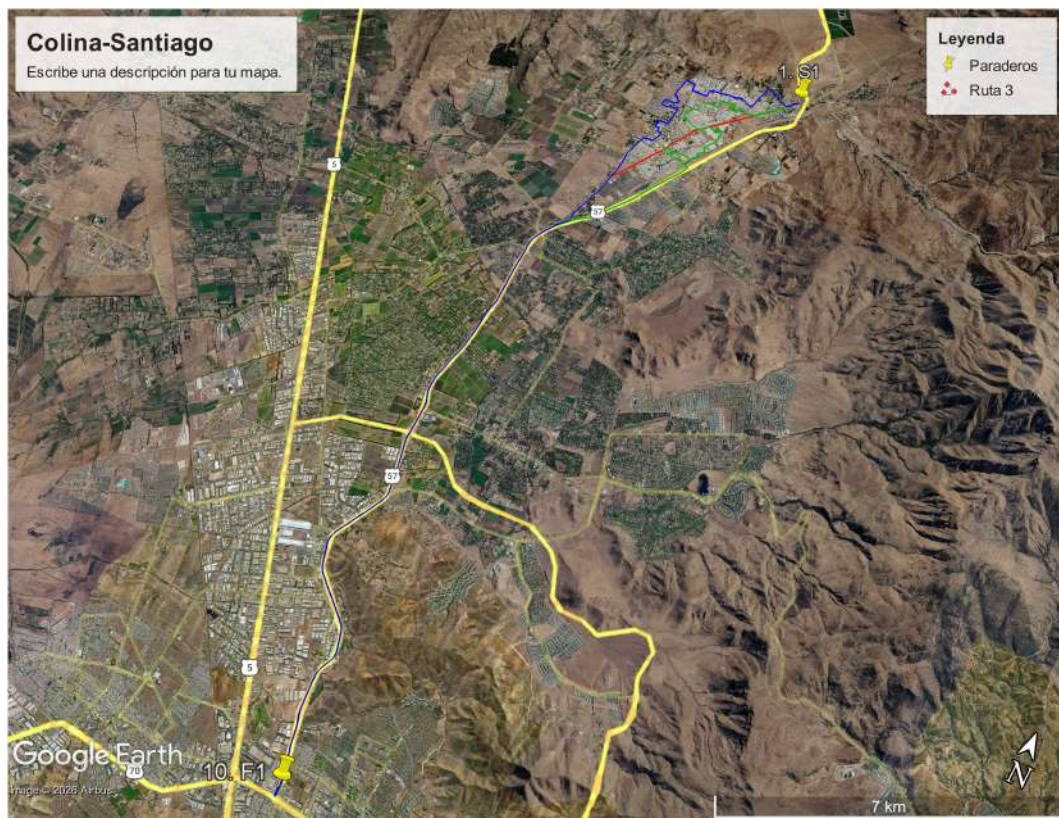
Imágen 11: Trazado generado para la Ruta 2, segunda iteración

La Ruta 2, mostrada en la Imagen 11, disminuyó de 26 a 20 paraderos tras la corrección, concentrando la mayor reducción de tiempo de las tres rutas. Su recorrido se mantiene en el sector nororiente y centro de Colina, utilizando el acceso E1. La magnitud de la reducción obtenida en esta ruta, 29,4 % del tiempo total, sugiere que varios de los paraderos removidos correspondían efectivamente a pasajes sin salida, y no solo a desvíos menores dentro del trazado.



Imágen 12: Trazado correspondiente a la Ruta 3, segunda iteración

La Ruta 3, presentada en la Imagen 12, conserva los mismos 7 paraderos de la ruta fija original y la misma secuencia de visita. La reducción de 2,25 minutos observada en esta ruta se explica por la reubicación de dos de sus paraderos hacia una posición más adecuada sobre la red vial, sin que ello haya alterado el orden del recorrido.



Imágen 13: Esquema conjunto de las rutas propuestas en la segunda iteración

La Imagen 13 permite visualizar de forma integrada las tres rutas corregidas. En comparación con el esquema de la primera iteración, los trazados presentan una menor cantidad de desvíos hacia el interior de las zonas residenciales, manteniendo la misma lógica de conexión hacia la Autopista Los Libertadores y la estación Los Libertadores.

Finalmente, la revisión del diagnóstico de pasajes sin salida aplicado sobre la nueva instancia no identificó paraderos que cumplieran simultáneamente los criterios de simetría y aislamiento utilizados para la detección automática. No obstante, dos paraderos presentaron un valor de aislamiento considerablemente superior al del resto de la instancia, por lo que se recomienda su revisión visual en una eventual tercera iteración del modelo.

5.6.1. Efecto de la corrección de pasajes sobre la cobertura de demanda

La corrección aplicada en la segunda iteración tiene una implicancia que no se limita al tiempo de las rutas: al retirar 8 paraderos del conjunto de candidatos y recalcular

la cobertura del modelo de localización sobre los 52 puntos restantes, la cobertura de demanda disminuye de forma considerable respecto de la solución original con 60 puntos.

Tabla 13: Efecto de la exclusión de paraderos sobre la cobertura de demanda

Indicador	60 paraderos (original)	52 paraderos (corregidos)
Cobertura R_1	79,48 %	67,72 %
Cobertura R_2	89,18 %	78,15 %
Nodos cubiertos R_1	798	674
Nodos cubiertos R_2	894	785

La exclusión de 8 de los 60 paraderos (13,3 % del total) reduce la cobertura en 11,76 % en R_1 y 11,03 % en R_2 , una disminución más que proporcional a la cantidad de puntos retirados. Esto evidencia un costo real de la corrección: los paraderos excluidos, si bien generaban recorridos vehiculares poco convenientes, también aportaban cobertura de demanda en sectores que, tras su exclusión, quedan sin ningún punto de recolección cercano.

Este resultado expone una tensión metodológica entre dos objetivos del sistema propuesto: por un lado, la factibilidad operacional de las rutas, que mejora sustancialmente al excluir los pasajes sin salida; por otro, la cobertura territorial de la demanda, que se ve afectada por esa misma exclusión. Una alternativa para mitigar esta pérdida sería incorporar puntos candidatos adicionales en las cercanías de los pasajes excluidos, pero ubicados sobre una vía con salida alternativa, de modo de recuperar cobertura en esos sectores sin reintroducir el problema operacional original. Esta posibilidad queda propuesta como trabajo futuro.

5.7. Discusión integrada de resultados

Los resultados del modelo de localización y del modelo de ruteo muestran que la propuesta permite avanzar desde un diagnóstico de cobertura hacia una estructura preliminar de operación. En primer lugar, la matriz de distancias permitió identificar que la cobertura máxima alcanzable estaba condicionada por la ubicación inicial de los puntos candidatos: el modelo de localización determinó que el escenario $p = 60$ entregaba el

mejor desempeño general, alcanzando exactamente ese límite geográfico, con coberturas de 79,48 % en el radio óptimo y 89,18 % en el radio deseable. En segundo lugar, el modelo de ruteo organizó estos 60 puntos en una estructura de tres rutas. Finalmente, la corrección de paraderos en pasajes sin salida, aplicada en una segunda iteración, redujo el conjunto a 52 puntos operacionalmente viables, con una cobertura de 67,72 % en R_1 y 78,15 % en R_2 (ver Tabla 13).

Respecto del estándar de referencia del 90 % adoptado del SIEDU/CNDU, ninguna de las dos soluciones lo alcanza. La cobertura máxima posible dentro del radio deseable, dada la ubicación de los 60 puntos candidatos originales, es de 89,18 %, es decir, un 0,82 % por debajo del estándar; tras la corrección de pasajes, la brecha se amplía a un 11,85 %. Esta diferencia no se origina en una limitación del procedimiento de optimización, sino en la disponibilidad geográfica de candidatos y, en la segunda iteración, en la necesidad de excluir puntos operacionalmente inviables. Alcanzar el 90 % requeriría incorporar nuevos puntos candidatos en los sectores actualmente sin cobertura.

La articulación entre ambas etapas muestra que la mejora del sistema no depende únicamente de aumentar el número de paraderos, sino de seleccionar ubicaciones adecuadas y luego integrarlas mediante recorridos operacionalmente razonables. Un conjunto de puntos bien localizado puede perder efectividad si no se organiza en rutas viables, mientras que un trazado eficiente pierde valor si no se relaciona correctamente con la distribución territorial de la demanda. El hallazgo de la sección anterior refuerza esta idea: mejorar la factibilidad operacional de las rutas tuvo un costo directo en cobertura, evidenciando que ambos objetivos deben evaluarse de forma conjunta y no de manera independiente.

Un aspecto relevante de la solución corresponde a la distinción entre rutas optimizadas y ruta fija. La Ruta 3 no fue generada por el modelo como una decisión de optimización, sino incorporada previamente por corresponder al trazado actualmente operativo en la comuna. Los paraderos restantes fueron optimizados y distribuidos en las Rutas 1 y 2, con tiempos totales de 53,50 y 48,96 minutos respectivamente tras la corrección espacial de la segunda iteración. Si bien ninguna de las dos rutas se encuentra cercana

al límite individual de 70 minutos, la suma de ambos tiempos continúa superando dicho límite, lo que explica que el modelo mantenga dos rutas optimizadas en lugar de consolidar todos los paraderos en un único recorrido.

Durante la inspección visual de los trazados de la primera iteración se observó un patrón en algunos sectores: ciertos paraderos podían inducir recorridos donde el vehículo debía ingresar y salir por la misma vía, generando una apariencia de desvío poco intuitiva. Para verificar si este comportamiento se debía a una limitación del procedimiento de solución, se realizó una prueba de control resolviendo nuevamente el modelo con la totalidad de los arcos posibles entre paraderos y exigiendo solución exacta. El resultado obtenido fue equivalente al de la formulación originalmente utilizada, lo que permite descartar que el comportamiento observado se debiera a una restricción artificial del espacio de búsqueda o a una solución subóptima.

La explicación de este comportamiento radica en la naturaleza del dato de entrada. El modelo opera sobre una matriz de tiempos origen-destino calculada entre puntos de interés, la cual representa una abstracción de la red vial real. Si bien OpenRouteService permite estimar tiempos de desplazamiento sobre la red vial, la formulación no modela explícitamente todos los atributos topológicos y operacionales de la red, como penalizaciones por giro, sentidos de circulación o maniobras asociadas a calles sin salida. Por ello, cuando un paradero se ubica al fondo de un pasaje, el tiempo de entrada y salida queda reflejado en la matriz, pero el trazado puede percibirse visualmente como poco natural desde una lectura cartográfica.

Frente a este hallazgo se aplicó una revisión diagnóstica y correctiva, desarrollada en dos etapas. En primer lugar, se implementó un diagnóstico automático que identifica paraderos sospechosos de encontrarse en pasajes sin salida, considerando simultáneamente la simetría entre los tiempos de entrada y salida y el aislamiento respecto de un segundo vecino cercano; este diagnóstico no reemplaza la revisión cartográfica, sino que acota los puntos a inspeccionar. En segundo lugar, se realizó una revisión visual directa de los trazados generados, identificando paraderos cuyo recorrido evidenciaba el patrón de

entrada y salida descrito. A partir de esta revisión, se corrigió la ubicación de 22 paraderos y se excluyeron 8, dando origen a la segunda iteración descrita en la sección anterior. Esta corrección no modifica el modelo de optimización, sino que mejora la calidad del dato geográfico utilizado como entrada.

Desde una perspectiva metodológica, el uso de OpenRouteService respondió principalmente a criterios de costo, acceso y reproducibilidad. Herramientas comerciales como Google Maps Platform podrían entregar información de red más detallada, incluyendo mejores atributos de circulación, penalización de giros y eventualmente condiciones de tráfico. Sin embargo, su uso requiere activar servicios de pago, lo que se aleja del alcance económico de este estudio. Por esta razón, se optó por OpenRouteService, una alternativa de acceso abierto basada en datos de OpenStreetMap, que permite construir una metodología replicable sin depender de licencias comerciales ni costos asociados al volumen de consultas.

Adicionalmente, debe considerarse que los tiempos utilizados en el modelo no incorporan condiciones dinámicas de tráfico vial. La matriz de tiempos fue construida a partir de estimaciones sobre la red vial, pero no considera congestión en horas punta, incidentes, variaciones por día de la semana ni demoras asociadas a la operación real en intersecciones. Por lo tanto, los tiempos obtenidos deben interpretarse como una aproximación base para comparar y construir rutas, y no como tiempos definitivos de operación. En la práctica, los tiempos de traslado podrían aumentar de manera significativa, especialmente en los tramos de conexión hacia la Autopista Los Libertadores y en el acceso a la estación Los Libertadores.

5.8. Comparación cuantitativa entre la situación actual y la propuesta

Con el fin de complementar la comparación espacial presentada anteriormente, se realizó una cuantificación directa entre la situación actual del sistema y la solución final obtenida, correspondiente a los 52 paraderos resultantes tras la segunda iteración.

Esta comparación se calculó íntegramente con distancias de red reales obtenidas mediante OpenRouteService para ambos casos, replicando sobre los 15 paraderos actualmente operativos el mismo procedimiento de cálculo de cobertura utilizado por el modelo de localización.

Tabla 14: Comparación cuantitativa entre la situación actual y la propuesta

Indicador	Situación actual	Propuesta final (52 paraderos)
Puntos de recolección	15	52
Cobertura R_1 (ponderada por demanda)	15,25 %	67,72 %
Cobertura R_2 (ponderada por demanda)	24,12 %	78,15 %
Nodos cubiertos R_1	157	674
Nodos cubiertos R_2	247	785

Como se observa en la Tabla 14, la propuesta mejora sustancialmente la cobertura de demanda respecto de la situación actual, tanto en el radio óptimo como en el radio deseable. Esta diferencia es considerablemente mayor a la estimada en un cálculo preliminar realizado con distancia en línea recta, lo que confirma que dicha aproximación subestimaba de forma importante la brecha real entre ambas situaciones, dado que no refleja el rodeo efectivo que exige la red vial de la comuna.

La magnitud de esta brecha se explica principalmente por la concentración de los paraderos actuales sobre un único corredor, la Caletera Oriente Gral. San Martín, lo que deja fuera de un radio caminable razonable a la mayor parte de los sectores residenciales interiores de Colina. La propuesta, al distribuir los puntos de recolección considerando explícitamente la ubicación de la demanda, logra una cobertura sustancialmente mayor incluso tras la reducción de 60 a 52 paraderos aplicada por motivos operacionales en la segunda iteración.

6. Conclusión y recomendaciones

6.1. Conclusión

El presente trabajo tuvo como objetivo diseñar una propuesta de mejora operacional para el sistema de transporte público interurbano entre Colina y Santiago, mediante la localización de puntos de recolección de pasajeros y la generación de rutas alimentadoras preliminares. A partir del desarrollo realizado, se logró construir una metodología integrada que combina análisis espacial, matrices de accesibilidad peatonal, optimización de cobertura y ruteo sobre red vial, permitiendo transformar una problemática territorial de acceso al transporte en una propuesta estructurada, evaluable y sometida a un proceso explícito de corrección y mejora.

En primer lugar, la construcción de la matriz de distancias permitió caracterizar la relación espacial entre los 991 nodos de demanda y los 70 puntos candidatos definidos en el área de estudio. Este insumo fue fundamental para identificar la cobertura potencial del sistema y evidenciar una restricción relevante, ya que con la configuración inicial de puntos candidatos, la cobertura máxima alcanzable corresponde a 79,48 % dentro del radio óptimo de 0,4 km y a 89,18 % dentro del radio deseable de 0,5 km. Por lo tanto, la imposibilidad de alcanzar el estándar de referencia del 90 % adoptado del SIEDU/CNDU no se debe al desempeño del modelo, sino a la ubicación geográfica de las alternativas disponibles, ya que 193 y 97 nodos de demanda, respectivamente, no cuentan con ningún candidato dentro de dichos radios.

En segundo lugar, el modelo de localización permitió evaluar seis escenarios según la cantidad de puntos de recolección a habilitar ($p = 20$ a $p = 70$). La cobertura resultó monótonamente creciente en p , con rendimientos marcadamente decrecientes, y el escenario $p = 60$ alcanzó exactamente el límite geográfico antes señalado (79,48 % y 89,18 %). El escenario $p = 70$ no aportó cobertura adicional alguna respecto de $p = 60$, confirmando que dicho límite es efectivamente el techo del sistema dado el conjunto de candidatos disponible. En consecuencia, $p = 60$ se adoptó como escenario base, por

representar la cobertura máxima posible con la menor cantidad de puntos habilitados.

La comparación entre la situación actual y la propuesta, realizada mediante distancias de red reales calculadas con OpenRouteService para ambos casos, mostró una diferencia sustancial. Los 15 paraderos actualmente operativos, concentrados sobre la Carretera Oriente General San Martín, cubren apenas 15,25 % de la demanda dentro del radio óptimo y 24,12 % dentro del radio deseable, mientras que la propuesta final del modelo de localización mejora estos valores de forma considerable. Esta comparación, calculada íntegramente sobre la red vial real, confirma que la concentración de la oferta actual sobre un único corredor deja a la mayoría de los sectores residenciales interiores de la comuna sin acceso peatonal razonable al sistema.

En relación con el ruteo, el proceso de diseño evidenció una limitación importante asociada a la herramienta de cálculo de tiempos utilizada. OpenRouteService permite estimar desplazamientos sobre la red vial real a costo nulo y con reproducibilidad metodológica, pero su abstracción no modela explícitamente penalizaciones por giro, sentidos de circulación ni la geometría de calles sin salida. Como consecuencia directa de esta limitación del motor de ruteo, la primera iteración de la solución generó trazados que, aunque matemáticamente óptimos respecto de la matriz de tiempos utilizada, resultaban operacionalmente poco convenientes en los sectores donde un paradero se ubicaba al fondo de un pasaje sin salida, obligando a un vehículo real a ingresar y salir por la misma vía.

Esta limitación fue identificada y corregida mediante un procedimiento de revisión en dos etapas. La primera consistió en un diagnóstico automático basado en la simetría de tiempos de entrada y salida y el aislamiento respecto de vías alternativas, y la segunda en una revisión visual directa de los trazados generados. A partir de esta revisión, se corrigió la ubicación de 22 paraderos y se excluyeron 8 casos sin alternativa de corrección, dando origen a una segunda iteración del modelo de ruteo sobre una instancia de 52 paraderos. Los resultados de esta segunda iteración confirman que la corrección fue efectiva, ya que el tiempo total de las rutas optimizadas se redujo entre 20,9 % y 29,4 %, una disminución más que proporcional a la cantidad de paraderos corregidos. Esto indica que los puntos

excluidos generaban un sobre costo de tiempo asociado específicamente al patrón de entrada y salida, y no una simple pérdida de cobertura marginal.

No obstante, esta corrección tuvo un costo medible. La cobertura de demanda del sistema final, evaluada sobre los 52 puntos operacionalmente viables, disminuye a 67,72 % en el radio óptimo y 78,15 % en el radio deseable, 11,76 y 11,03 puntos porcentuales por debajo de la solución de 60 puntos. Este hallazgo expone una tensión real entre dos objetivos del sistema propuesto, la factibilidad operacional de las rutas y la cobertura territorial de la demanda, que no puede resolverse dentro del alcance de este trabajo, y que se traduce en la recomendación de incorporar puntos candidatos adicionales en los sectores afectados.

La propuesta final quedó estructurada en tres rutas, dos optimizadas (53,50 y 48,96 minutos, con 25 y 20 paraderos respectivamente) y una ruta fija correspondiente al trazado actualmente operativo (26,40 minutos, 7 paraderos), conectadas mediante dos accesos a la Autopista Los Libertadores hacia la estación intermodal del mismo nombre. El modelo formulado corresponde a una variante del problema de ruteo de vehículos (VRP) con restricción de duración por ruta, resuelto para el sentido de viaje Colina-Santiago. El recorrido de regreso se asume equivalente a la misma ruta en sentido inverso.

Los resultados obtenidos deben interpretarse como una propuesta técnica preliminar. El modelo no incorpora condiciones dinámicas de tráfico vial, penalizaciones de giro ni sentidos de circulación, por lo que los tiempos estimados representan una base de comparación entre alternativas y no tiempos definitivos de operación. Asimismo, tanto la demanda por nodo como la capacidad de los puntos candidatos se estimaron mediante supuestos metodológicos explícitos y verificables, dado que no se dispone de mediciones desagregadas reales y esa información solo está disponible para el transporte público licitado. La solidez de dichos supuestos fue verificada mediante análisis de robustez multi-semilla.

En síntesis, la metodología propuesta permite abordar de manera coherente y verificable el problema de accesibilidad al transporte público interurbano entre Colina y

Santiago. El modelo de localización identifica puntos de recolección significativamente más cercanos a la demanda que la situación actual, mientras que el modelo de ruteo, tras un proceso explícito de diagnóstico y corrección, organiza dichos puntos en una estructura operacionalmente viable. La propuesta constituye una base técnica sólida para una etapa posterior de validación operacional, estimación de frecuencias y dimensionamiento de flota, cuyas líneas específicas se detallan en las recomendaciones siguientes.

6.2. Recomendaciones

A partir de los resultados obtenidos, se recomienda mantener el escenario $p = 60$ como base del diseño preliminar del sistema, por representar la cobertura máxima alcanzable dado el conjunto de puntos candidatos disponible, sin que un mayor número de puntos habilitados aporte cobertura adicional.

Se recomienda ampliar el conjunto de puntos candidatos en los sectores que permanecen fuera de cobertura, así como en las proximidades de los pasajes excluidos durante la segunda iteración del ruteo. La incorporación de nuevas alternativas espaciales sobre vías con salida alternativa permitiría recuperar parte de la cobertura perdida en la corrección operacional, sin reintroducir el problema de trazados poco convenientes.

Respecto del ruteo, se recomienda adoptar la Ruta 1 y la Ruta 2 de la segunda iteración como propuestas alimentadoras preliminares, manteniendo la Ruta 3 como trazado fijo asociado al recorrido actualmente operativo. Se recomienda además incorporar el procedimiento de diagnóstico y revisión visual de pasajes sin salida como una etapa estándar del proceso de diseño, dado que su aplicación demostró un efecto significativo sobre la calidad operacional de las rutas.

Como líneas de trabajo futuro, orientadas a un estudio de continuidad que exceda el alcance metodológico y de recursos de este trabajo, se identifican las siguientes.

- **Mejorar la estimación de demanda por nodo**, reemplazando la asignación mediante distribución uniforme por información desagregada real, como encuestas de

origen-destino, catastros de vivienda o densidad poblacional por manzana, en la medida en que dicha información se encuentre disponible para el transporte privado y los modos no licitados.

- **Mejorar el motor de cálculo de rutas**, evaluando alternativas comerciales como Google Maps Platform o herramientas de código abierto con topología vial completa, como OSMnx, que permitan modelar explícitamente penalizaciones de giro, sentidos de circulación y condiciones de tráfico en tiempo real.
- **Incorporar restricciones operacionales adicionales** al modelo de ruteo, tales como capacidad de carga por vehículo, frecuencia de servicio y disponibilidad de flota, en lugar de la restricción de duración utilizada en este trabajo.
- **Incorporar el estado real de la red vial**, incluyendo obras en curso, restricciones de circulación y condiciones de congestión por horario, de manera de acercar los tiempos estimados a la operación efectiva del sistema.

Finalmente, se recomienda complementar la propuesta con una etapa de dimensionamiento operacional que considere frecuencias, cantidad de buses requeridos, tiempos de ciclo, capacidad de transporte y horarios de operación, además de una validación en terreno de los tiempos de viaje estimados, especialmente en los tramos de conexión hacia la Autopista Los Libertadores y el acceso a la estación Los Libertadores. De esta manera, los resultados del modelo de localización y del modelo de ruteo podrían transformarse en una propuesta robusta y validada para la mejora del sistema de transporte público interurbano entre Colina y Santiago.

7. Referencias

- Biblioteca del Congreso Nacional de Chile. (2024). *Reporte comunal de Colina*. Santiago, Chile. Recuperado de <https://www.bcn.cl/siit/reportescomunales/reporpdf.html?anno=2024&idcom=13301>
- Chicureo Hoy. (2022, mayo 18). *Sondeo sobre transporte público en Colina: un 63,9 % de los vecinos se moviliza en sus vehículos*. Recuperado de <https://www.chicureo.hoy.cl/actualidad/sondeo-sobre-transporte-publico-en-colina-un-639-de-los-vecinos-se-moviliza-en-sus-vehiculos/>
- Martí, P., Jordán, J., González Arrieta, A., & Julián, V. (2023). *A Survey on Demand-Responsive Transportation for Rural and Interurban Mobility*. *International Journal of Interactive Multimedia and Artificial Intelligence*, 8(3), 43–54. <https://doi.org/10.9781/ijimai.2023.07.010>
- Ministerio de Transportes y Telecomunicaciones (MTT). (2013). *Plan Maestro de Transporte Santiago 2025*. Santiago, Chile: Gobierno de Chile.
- Municipalidad de Colina. (2021). *Plan de Desarrollo Comunal 2020-2030: Resumen ejecutivo*. Colina, Chile.
- Sociedad Chilena de Ingeniería de Transporte (SOCHITRAN). (2013). *Políticas de transporte urbano para nuestras ciudades*. Santiago, Chile.
- Ballou, R. H. (2004). *Logística: Administración de la cadena de suministro* (5.a ed.). México: Pearson Educación.
- Church, R., & ReVelle, C. (1974). The maximal covering location problem. *Papers of the Regional Science Association*, 32(1), 101–118.
- Current, J., & Schilling, D. (1990). Analysis of errors due to demand data aggregation in

- the set covering and maximal covering location problems. *Geographical Analysis*, 22(2), 116–126. <https://doi.org/10.1111/j.1538-4632.1990.tb00199.x>
- Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1), 80–91. <https://doi.org/10.1287/mnsc.6.1.80>
- Dantzig, G. B., Fulkerson, D. R., & Johnson, S. M. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4), 393–410.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271. <https://doi.org/10.1007/BF01386390>
- Hakimi, S. L. (1964). Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12(3), 450–459. <https://doi.org/10.1287/opre.12.3.450>
- Hitchcock, F. L. (1941). The distribution of a product from several sources to numerous localities. *Journal of Mathematics and Physics*, 20(1–4), 224–230. <https://doi.org/10.1002/sapm1941201224>
- Koopmans, T. C. (1949). Optimum utilization of the transportation system. *Econometrica*, 17, 136–146.
- Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. Philadelphia, PA: Society for Industrial and Applied Mathematics.
- Chicureo Hoy. (2026, marzo 4). *Colina anotó el mayor número de venta de casas nuevas de la RM en 2025*. Recuperado de <https://www.chicureohoy.cl/actualidad/colina-anoto-el-mayor-numero-de-venta-de-casas-nuevas-de-la-rm-en-2025/>
- Ministerio de Desarrollo Social y Familia. (2024). *Instructivo metodológico para la incorporación del hacinamiento en la evaluación social de proyectos de transporte*

- público mayor* (Edición 2). Sistema Nacional de Inversiones, Gobierno de Chile.
- Ministerio de Transportes y Telecomunicaciones. (2025). *Reglamento de los servicios de transporte público remunerado de pasajeros* [Documento en consulta pública]. Gobierno de Chile.
- Ministerio de Transportes y Telecomunicaciones. (2004). *Decreto Supremo N.º 80: Reglamenta transporte privado remunerado de pasajeros*. Gobierno de Chile.
- Ministerio de Transportes y Telecomunicaciones. (1991). *Decreto Supremo N.º 122: Fija requisitos dimensionales y de otras características técnicas para vehículos que indica*. Gobierno de Chile.
- Ministerio de Vivienda y Urbanismo. *Ordenanza General de Urbanismo y Construcciones (OGUC)*. Título 2, Capítulo 3, Artículo 2.3.2.
- Instituto Nacional de Estadísticas. (2020). *Manual de metadatos SIEDU: Línea de Base 2018*. Sistema de Indicadores y Estándares de Desarrollo Urbano.
- Instituto Nacional de Estadísticas. (2019). *Sistema de Indicadores y Estándares del Desarrollo Urbano (SIEDU): Indicador BPU_25 — Porcentaje de la población que accede a paraderos de transporte público*. Ministerio de Desarrollo Social y Familia.
- Consejo Nacional de Desarrollo Urbano. (2018). *Propuesta de Sistema de Indicadores y Estándares de Desarrollo Urbano*. Consejo Nacional de Desarrollo Urbano.
- Neis, P., & Zipf, A. (2008). *OpenRouteService.org is three times .open": Combining OpenSource, OpenLS and OpenStreetMaps*. GIS Research UK (GISRUK 08). Manchester, Reino Unido.
- Haklay, M., & Weber, P. (2008). *OpenStreetMap: User-generated street maps*. IEEE Pervasive Computing, 7(4), 12-18. <https://doi.org/10.1109/MPRV.2008.80>

8. Anexos

8.1. Código para la construcción de la matriz de distancias y tiempos

El siguiente código permite importar la base georreferenciada construida en Google Earth Pro, extraer las coordenadas de los nodos de demanda y puntos candidatos de recolección, calcular las matrices de distancia y tiempo de caminata mediante OpenRouteService, y exportar los resultados en formato JSON.

```
1 import os
2 import json
3 import geopandas as gpd
4 import pandas as pd
5 import numpy as np
6 import openrouteservice
7 # -----
8 # 1. Configuración inicial
9 # -----
10 kml_path = "Mis_lugares.kmz"
11 api_key = "TU_API_KEY"
12 client = openrouteservice.Client(key=api_key)
13 # -----
14 # 2. Lectura del archivo KMZ
15 # -----
16 Paraderos = gpd.read_file(kml_path, driver="KMZ", layer="Candidatos")
17 demandas = gpd.read_file(kml_path, driver="KMZ", layer="Demanda")
18 # -----
19 # 3. Extracción de coordenadas
20 # -----
21 Paraderos["lat"] = Paraderos.geometry.y
22 Paraderos["lon"] = Paraderos.geometry.x
23
24 demandas["lat"] = demandas.geometry.y
```

```

25 demandas["lon"] = demandas.geometry.x
26 # -----
27 # 4. Creacion de tablas limpias
28 # -----
29 df_paraderos = Paraderos[["Name", "lat", "lon"]].reset_index(drop=True)
30 df_demandas = demandas[["Name", "lat", "lon"]].reset_index(drop=True)
31
32 df_paraderos["id"] = range(1, len(df_paraderos) + 1)
33 df_demandas["id"] = range(1, len(df_demandas) + 1)
34
35 def limpiar_df(df, tipo):
36     df = df.rename(columns=lambda x: x.strip().lower())
37     df = df[["id", "name", "lat", "lon"]]
38     print(f"{tipo} limpiado: {len(df)} filas")
39     return df.reset_index(drop=True)
40
41 df_paraderos = limpiar_df(df_paraderos, "Centros")
42 df_demandas = limpiar_df(df_demandas, "Demandas")
43
44 print("----- Paraderos -----")
45 print(df_paraderos.head())
46
47 print("----- Demandas -----")
48 print(df_demandas.head())
49 # -----
50 # 5. Preparacion de coordenadas para OpenRouteService
51 # -----
52 coords_paraderos = list(zip(df_paraderos["lon"], df_paraderos["lat"]))
53 coords_demandas = list(zip(df_demandas["lon"], df_demandas["lat"]))
54
55 n_centros = len(coords_paraderos)
56 n_demandas = len(coords_demandas)
57 # -----
58 # 6. Calculo de matriz de distancias y tiempos

```

```

59 # -----
60 max_batch = 50
61
62 distances = []
63 times = []
64
65 for start in range(0, n_demandas, max_batch):
66     end = min(start + max_batch, n_demandas)
67
68     print(f"Procesando demandas {start + 1} a {end} de {n_demandas}")
69
70     sub_coords_demandas = coords_demandas[start:end]
71
72     matrix = client.distance_matrix(
73         locations=coords_paraderos + sub_coords_demandas,
74         sources=list(range(n_centros, n_centros + len(
75             sub_coords_demandas))),
76         destinations=list(range(n_centros)),
77         profile="foot-walking",
78         metrics=["distance", "duration"]
79     )
80
81     distances.append(np.array(matrix["distances"]) / 1000)
82     times.append(np.array(matrix["durations"]) / 60)
83
84 distances = np.vstack(distances)
85 times = np.vstack(times)
86
87 # -----
88 # 7. Crear matrices como DataFrame
89 # -----
90 df_distancias = pd.DataFrame(
91     distances,
92     index=df_demandas["id"],
93     columns=df_paraderos["id"]

```

```

92 )
93
94 df_tiempos = pd.DataFrame(
95     times,
96     index=df_demandas["id"],
97     columns=df_paraderos["id"]
98 )
99 # -----
100 # 8. Crear estructura JSON
101 # -----
102 resultado_json = {
103     "descripcion": {
104         "archivo_origen": kml_path,
105         "perfil_ruta": "foot-walking",
106         "unidad_distancia": "kilometros",
107         "unidad_tiempo": "minutos",
108         "cantidad_puntos_candidatos": int(n_centros),
109         "cantidad_nodos_demanda": int(n_demandas)
110     },
111     "puntos_candidatos": df_paraderos.to_dict(orient="records"),
112     "nodos_demanda": df_demandas.to_dict(orient="records"),
113     "matriz_distancias_km": df_distancias.to_dict(orient="index"),
114     "matriz_tiempos_min": df_tiempos.to_dict(orient="index")
115 }
116 # -----
117 # 9. Exportar archivo JSON
118 # -----
119 output_json = "resultado_matriz_colina.json"
120 with open(output_json, "w", encoding="utf-8") as f:
121     json.dump(resultado_json, f, ensure_ascii=False, indent=4)
122 print("Archivo JSON generado correctamente:")
123 print(output_json)

```

8.2. Código para resolver el modelo de localización

En el presente anexo se muestra el código utilizado para implementar el modelo de localización en Python mediante la librería Pyomo. El procedimiento considera la carga de la matriz de distancias caminables, la generación de parámetros del modelo, la construcción de las vecindades de cobertura, la definición de variables, restricciones y función objetivo, y la resolución de los escenarios asociados a distintos valores de p .

```
1 import json
2 import shutil
3 import numpy as np
4 import pandas as pd
5 from pyomo.environ import (
6     ConcreteModel, Set, Param, Var, Binary, NonNegativeReals,
7     Constraint, Objective, SolverFactory, value, maximize
8 )
9 # Parametros generales
10 SEED = 3
11 np.random.seed(SEED)
12 archivo_base = "resultado_matriz_colina.json"
13 R1 = 0.4
14 R2 = 0.5
15 O1 = 2
16 H_actual = 1973.33
17 gamma = 0.50
18 H_C = H_actual * (1 + gamma)
19 p_escenarios = [20, 30, 40, 50, 60, 70]
20 cobertura_minima = None
21 capacidad_bus = 90
22 buses_por_hora = 2
23 duracion_periodo_horas = 5
24 capacidad_por_punto = capacidad_bus * buses_por_hora *
    duracion_periodo_horas
```

```

25 solver_name = "cbc"
26 tiempo_maximo_solver = 180
27 gap_mip = 0.01
28 # Carga de datos
29 with open(archivo_base, "r", encoding="utf-8") as f:
30     datos = json.load(f)
31 df_candidatos = pd.DataFrame(datos["puntos_candidatos"]).sort_values("
    id").reset_index(drop=True)
32 df_demanda = pd.DataFrame(datos["nodos_demanda"]).sort_values("id").
    reset_index(drop=True)
33 df_dist = pd.DataFrame(datos["matriz_distancias_km"])
34 if df_dist.shape[0] == len(df_candidatos) and df_dist.shape[1] == len(
    df_demanda):
35     df_dist = df_dist.T
36 df_dist.index = df_dist.index.astype(str)
37 df_dist.columns = df_dist.columns.astype(str)
38 ids_demanda = df_demanda["id"].astype(str).tolist()
39 ids_candidatos = df_candidatos["id"].astype(str).tolist()
40 dist = df_dist.loc[ids_demanda, ids_candidatos].astype(float).values
41 n_i, n_j = dist.shape
42 # Generacion de demanda
43 limite_superior_h = 2 * H_C / n_i
44 h = np.random.uniform(0, limite_superior_h, size=n_i)
45 # Construccion de vecindades
46 N1 = {i: [j for j in range(n_j) if dist[i, j] <= R1] for i in range(n_i
    )}
47 N2 = {i: [j for j in range(n_j) if dist[i, j] <= R2] for i in range(n_i
    )}
48
49 pares_N1 = [(i, j) for i in range(n_i) for j in N1[i]]
50 pares_N2 = [(i, j) for i in range(n_i) for j in N2[i]]
51 # Funcion para obtener solver
52 def obtener_solver():
53     cbc_path = shutil.which("cbc") or shutil.which("cbc.exe")

```

```

54 glpk_path = shutil.which("glpsol") or shutil.which("glpsol.exe")
55
56 if cbc_path is not None:
57     return SolverFactory("cbc", executable=cbc_path), "cbc"
58 elif glpk_path is not None:
59     return SolverFactory("glpk", executable=glpk_path), "glpk"
60 else:
61     raise RuntimeError("No se encontro CBC ni GLPK.")
62
63
64 # Funcion para resolver cada escenario
65 def resolver_escenario(p_valor):
66     b = np.full(n_j, capacidad_por_punto, dtype=float)
67     model = ConcreteModel()
68     # Conjuntos
69     model.I = Set(initialize=list(range(n_i)))
70     model.J = Set(initialize=list(range(n_j)))
71     model.P2 = Set(initialize=pares_N2, dimen=2)
72
73     # Parametros
74     model.h = Param(
75         model.I,
76         initialize={i: float(h[i]) for i in range(n_i)},
77         within=NonNegativeReals
78     )
79     model.b = Param(
80         model.J,
81         initialize={j: float(b[j]) for j in range(n_j)},
82         within=NonNegativeReals
83     )
84     # Variables
85     model.X = Var(model.J, domain=Binary)
86     model.Y = Var(model.P2, domain=Binary)
87     model.Z = Var(model.I, domain=Binary)

```

```

88 model.W = Var(model.I, domain=Binary)
89 # Cobertura optima
90 def cobertura_optima_rule(m, i):
91     if len(N1[i]) == 0:
92         return m.Z[i] == 0
93     return m.Z[i] <= sum(m.Y[i, j] for j in N1[i])
94
95 model.CoberturaOptima = Constraint(model.I, rule=
    cobertura_optima_rule)
96 # Cobertura deseable y asignacion unica
97 def cobertura_deseable_rule(m, i):
98     if len(N2[i]) == 0:
99         return m.W[i] == 0
100     return sum(m.Y[i, j] for j in N2[i]) == m.W[i]
101
102 model.CoberturaDeseable = Constraint(model.I, rule=
    cobertura_deseable_rule)
103 # Relacion entre cobertura optima y deseable
104 def relacion_cobertura_rule(m, i):
105     return m.Z[i] <= m.W[i]
106
107 model.RelacionCobertura = Constraint(model.I, rule=
    relacion_cobertura_rule)
108
109 # Vinculacion entre asignacion y apertura
110 def vinculacion_rule(m, i, j):
111     return m.Y[i, j] <= m.X[j]
112
113 model.Vinculacion = Constraint(model.P2, rule=vinculacion_rule)
114 # Capacidad maxima
115 def capacidad_rule(m, j):
116     nodos = [i for i in range(n_i) if j in N2[i]]
117     if len(nodos) == 0:
118         return Constraint.Skip

```

```

119         return sum(m.h[i] * m.Y[i, j] for i in nodos) <= m.b[j] * m.X[j
120     ]
121
122 model.Capacidad = Constraint(model.J, rule=capacidad_rule)
123
124 # Numero de puntos a seleccionar
125 def seleccion_p_rule(m):
126     return sum(m.X[j] for j in m.J) == p_valor
127
128 model.SeleccionP = Constraint(rule=seleccion_p_rule)
129
130 # Cobertura minima, si se activa
131 if cobertura_minima is not None:
132     def cobertura_minima_rule(m):
133         return sum(m.h[i] * m.W[i] for i in m.I) >=
134             cobertura_minima * sum(m.h[i] for i in m.I)
135
136 model.CoberturaMinima = Constraint(rule=cobertura_minima_rule)
137
138 # Funcion objetivo
139 def objetivo_rule(m):
140     return (
141         01 * sum(m.h[i] * m.Z[i] for i in m.I)
142         + sum(m.h[i] * m.W[i] for i in m.I)
143     )
144
145 model.Obj = Objective(rule=objetivo_rule, sense=maximize)
146
147 # Resolucion
148 solver, solver_usado = obtener_solver()
149
150 if solver_usado == "cbc":
151     solver.options["seconds"] = tiempo_maximo_solver
152     solver.options["ratioGap"] = gap_mip
153
154 if solver_usado == "glpk":

```

```

151     solver.options["tmlim"] = tiempo_maximo_solver
152     solver.options["mipgap"] = gap_mip
153
154     resultado = solver.solve(model, tee=False)
155     X_val = {j: int(round(value(model.X[j]))) for j in model.J}
156     Z_val = {i: int(round(value(model.Z[i]))) for i in model.I}
157     W_val = {i: int(round(value(model.W[i]))) for i in model.I}
158     demanda_total = sum(h)
159     demanda_R1 = sum(h[i] * Z_val[i] for i in range(n_i))
160     demanda_R2 = sum(h[i] * W_val[i] for i in range(n_i))
161     puntos_seleccionados = [j for j, val in X_val.items() if val == 1]
162     return {
163         "p": p_valor,
164         "valor_objetivo": value(model.Obj),
165         "cobertura_R1": 100 * demanda_R1 / demanda_total,
166         "cobertura_R2": 100 * demanda_R2 / demanda_total,
167         "nodos_R1": sum(Z_val.values()),
168         "nodos_R2": sum(W_val.values()),
169         "puntos_seleccionados": puntos_seleccionados
170     }
171 # Resolucion de escenarios
172 resultados = []
173 for p_valor in p_escenarios:
174     resultados.append(resolver_escenario(p_valor))
175 df_resultados = pd.DataFrame(resultados)
176 print(df_resultados)

```

8.3. Código para resolver el modelo de ruteo

En el presente anexo se muestra el código utilizado para implementar el modelo de ruteo en Python mediante la librería Pyomo. El procedimiento considera la carga de puntos georreferenciados, el cálculo de tiempos sobre red vial mediante OpenRouteService, la

depuración de paraderos ubicados en pasajes sin salida, la construcción de arcos factibles, la definición de variables, restricciones y función objetivo, y la resolución del modelo minimizando el tiempo total de las rutas generadas.

```
1 import os,time,zipfile,random,xml.etree.ElementTree as ET
2 from collections import deque
3 import pandas as pd
4 import pyomo.environ as pyo
5 import openrouteservice,requests
6
7 KML_FILE="Modelo resuelto.kmz"
8 ORS_API_KEY=os.getenv("ORS_API_KEY","PEGA_AQUI_TU_API_KEY")
9 PROFILE="driving-car"
10 TMAX=70
11 TIEMPO_DETENCION_PARADERO=0.5
12 USAR_SNAP=True
13 SNAP_RADIUS_M=350
14 NUM_ZONAS=None
15 RUTAS_EXTRA_POR_ZONA=4
16 KMAX_SIN_ZONAS=6
17 K_VECINOS_ORIGEN=12
18 K_VECINOS_PARADERO=14
19 K_VECINOS_ENTRADA=8
20 K_VECINOS_ACCESO=4
21 MAX_TIEMPO_ARCO_PP=25
22 TIME_LIMIT_SEG=45
23 MIP_GAP=0.02
24 MAX_ITER_CORTES=25
25 OMITIR_PASAJES_SIN_SALIDA=True
26 MAX_PORCENTAJE_OMISION=0.25
27 LIMITE_RUTAS_ORS=3500
28 RUTA_FIJA_NOMBRES=["S1","P24","P53","P69","P19","P11","P12","P4","E2"]
29
```

```

30 if not ORS_API_KEY or ORS_API_KEY=="PEGA_AQUI_TU_API_KEY":
31     raise ValueError("Ingresar API key de OpenRouteService.")
32 client=openrouteservice.Client(key=ORS_API_KEY)
33
34 def leer_kml_o_kmz(path):
35     if path.lower().endswith(".kmz"):
36         with zipfile.ZipFile(path,"r") as z:
37             kmls=[n for n in z.namelist() if n.lower().endswith(".kml")]
38             data=z.read(kmls[0])
39     else:
40         data=open(path,"rb").read()
41     root=ET.fromstring(data)
42     ns={"kml":"http://www.opengis.net/kml/2.2"}
43     pts=[]
44     for pm in root.findall("./kml:Placemark",ns):
45         ne=pm.find("kml:name",ns)
46         ce=pm.find("./kml:coordinates",ns)
47         if ne is None or ce is None:
48             continue
49         lon,lat,*_=ce.text.strip().split()[0].split(",")
50         pts.append({"nombre":ne.text.strip(),"lat":float(lat),"lon":
51                     float(lon)})
52     return pd.DataFrame(pts)
53
54 def detectar_tipo(nombre):
55     n=nombre.upper().strip()
56     if n.startswith("S"):
57         return "ORIGEN"
58     if n.startswith("F"):
59         return "DESTINO"
60     if n.startswith("E"):
61         return "ACCESO"
62     if n.startswith("P"):

```

```

62         return "PARADERO"
63     return "OTRO"
64
65 def snap_points_ors(df):
66     df=df.copy().reset_index(drop=True)
67     url=f"https://api.openrouteservice.org/v2/snap/{PROFILE}/json"
68     body={"locations":df[["lon","lat"]].values.tolist(),"radius":
        SNAP_RADIUS_M}
69     r=requests.post(
70         url,
71         json=body,
72         headers={"Authorization":ORS_API_KEY,"Content-Type":"
            application/json"},
73         timeout=90
74     )
75     r.raise_for_status()
76     for k,item in enumerate(r.json().get("locations",[])):
77         if item is not None:
78             df.loc[k,"lon"],df.loc[k,"lat"]=item["location"]
79     return df
80
81 def validar_datos(df):
82     if len(df[df.tipo=="ORIGEN"])!=1:
83         raise ValueError("Debe existir un origen S.")
84     if len(df[df.tipo=="DESTINO"])!=1:
85         raise ValueError("Debe existir un destino F.")
86     if len(df[df.tipo=="ACCESO"])==0:
87         raise ValueError("Debe existir al menos un acceso E.")
88     if len(df[df.tipo=="PARADERO"])==0:
89         raise ValueError("Debe existir al menos un paradero P.")
90
91 df=leer_kml_o_kmz(KML_FILE)
92 df["tipo"]=df["nombre"].apply(detector_tipo)
93 validar_datos(df)

```

```

94
95 if USAR_SNAP:
96     df=snap_points_ors(df)
97
98     origen="0"
99     destino="F"
100     origen_row=df[df.tipo=="ORIGEN"].iloc[0]
101     destino_row=df[df.tipo=="DESTINO"].iloc[0]
102     paraderos_df=df[df.tipo=="PARADERO"].copy().reset_index(drop=True)
103     accesos_df=df[df.tipo=="ACCESO"].copy().reset_index(drop=True)
104
105     P=[f"P{i}" for i in range(len(paraderos_df))]
106     E=[f"E{i}" for i in range(len(accesos_df))]
107     D=P+E
108     N=[origen]+D+[destino]
109
110     coord={
111         origen:(origen_row.lat,origen_row.lon),
112         destino:(destino_row.lat,destino_row.lon)
113     }
114     nombre={
115         origen:origen_row.nombre,
116         destino:destino_row.nombre
117     }
118     tipo_nodo={
119         origen:"ORIGEN",
120         destino:"DESTINO"
121     }
122
123     for k,row in paraderos_df.iterrows():
124         n=f"P{k}"
125         coord[n]=(row.lat,row.lon)
126         nombre[n]=row.nombre
127         tipo_nodo[n]="PARADERO"

```

```

128
129 for k,row in accesos_df.iterrows():
130     n=f"E{k}"
131     coord[n]=(row.lat,row.lon)
132     nombre[n]=row.nombre
133     tipo_nodo[n]="ACCESO"
134
135 A_base=[]
136 for i in N:
137     for j in N:
138         if i==j or j==origen or i==destino:
139             continue
140         if i==origen and j not in P:
141             continue
142         if i in P and j not in P+E:
143             continue
144         if i in E and j!=destino:
145             continue
146         if j==destino and i not in E:
147             continue
148         A_base.append((i,j))
149
150 def calcular_matriz_ors():
151     locations=[[coord[n][1],coord[n][0]] for n in N]
152     total_n=len(locations)
153     tamaño_lote=max(1,LIMITE_RUTAS_ORS//total_n)
154
155     durations=[[None]*total_n for _ in range(total_n)]
156     distances=[[None]*total_n for _ in range(total_n)]
157     indices_destino=list(range(total_n))
158
159     for inicio in range(0,total_n,tamaño_lote):
160         fin=min(inicio+tamaño_lote,total_n)
161         indices_origen=list(range(inicio,fin))

```

```

162     matrix=client.distance_matrix(
163         locations=locations,
164         sources=indices_origen,
165         destinations=indices_destino,
166         profile=PROFILE,
167         metrics=["duration","distance"],
168         units="km"
169     )
170     for offset,fila_origen in enumerate(indices_origen):
171         durations[fila_origen]=matrix["durations"][offset]
172         distances[fila_origen]=matrix["distances"][offset]
173
174     idx={n:k for k,n in enumerate(N)}
175     filas=[]
176     for i,j in A_base:
177         a,b=idx[i],idx[j]
178         d=durations[a][b]
179         km=distances[a][b]
180         if d is not None and km is not None:
181             filas.append({
182                 "i":i,
183                 "j":j,
184                 "tiempo_min":d/60,
185                 "distancia_km":km
186             })
187     return pd.DataFrame(filas)
188
189 matriz_df=calcular_matriz_ors()
190 A_base_set=set(A_base)
191 t_all={
192     (r.i,r.j):float(r.tiempo_min)
193     for r in matriz_df.itertuples()
194     if (r.i,r.j) in A_base_set
195 }

```

```

196 distancia_all={
197     (r.i,r.j):float(r.distancia_km)
198     for r in matriz_df.itertuples()
199     if (r.i,r.j) in A_base_set
200 }
201
202 def diagnosticar_pasajes():
203     filas=[]
204     for i in P:
205         vec=sorted(
206             [(j,t_all[(i,j)]) for j in P if i!=j and (i,j) in t_all],
207             key=lambda x:x[1]
208         )
209         if len(vec)<2:
210             continue
211         j1,t1=vec[0]
212         t2=vec[1][1]
213         tv=t_all.get((j1,i))
214         if tv is None:
215             continue
216         similitud=max(0,1-abs(t1-tv)/max(t1,tv,0.01))
217         aislamiento=(t2-t1)/max(t1,0.01)
218         puntaje=similitud*0.45+min(aislamiento,3)/3*0.55
219         filas.append({
220             "nodo":i,
221             "similitud":similitud,
222             "aislamiento":aislamiento,
223             "tiempo_vecino":t1,
224             "puntaje":puntaje
225         })
226     return pd.DataFrame(filas).sort_values("puntaje",ascending=False)
227
228 def seleccionar_omitidos(diag):
229     if not OMITIR_PASAJES_SIN_SALIDA or diag.empty:

```

```

230         return set()
231     c=diag[
232         (diag.similitud>=0.75)&
233         (diag.aislamiento>=1.50)&
234         (diag.tiempo_vecino<=5.0)
235     ]
236     max_om=int(len(P)*MAX_PORCENTAJE_OMISION)
237     if max_om>0 and len(c)>max_om:
238         c=c.head(max_om)
239     return set(c.nodo.tolist())
240
241 omitidos=seleccionar_omitidos(diagnosticar_pasajes())
242 P=[p for p in P if p not in omitidos]
243
244 def cargar_ruta_fija():
245     mapa={v.strip().upper():k for k,v in nombre.items()}
246     ruta=[]
247     for nm in RUTA_FIJA_NOMBRES:
248         ruta.append(mapa[nm.strip().upper()])
249     if ruta[0]!=origen:
250         ruta=[origen]+ruta
251     if ruta[-1]!=destino:
252         ruta.append(destino)
253     for i,j in zip(ruta[:-1],ruta[1:]):
254         if (i,j) not in t_all:
255             raise RuntimeError("La ruta fija contiene tramos sin tiempo
256                                ORS.")
257     return [("FIJA-R1",ruta)]
258
259 rutas_fijas_resultado=cargar_ruta_fija()
260
261 for _,ruta in rutas_fijas_resultado:
262     P=[p for p in P if p not in ruta]

```

```

263 zonas={0:list(P)}
264
265 def construir_arcos_zona(Pz):
266     A=set()
267     for j,_ in sorted(
268         [(j,t_all[(origen,j)]) for j in Pz if (origen,j) in t_all],
269         key=lambda x:x[1]
270    )[:K_VECINOS_ORIGEN]:
271         A.add((origen,j))
272
273     for i in Pz:
274         vecinos=sorted(
275             [(j,t_all[(i,j)]) for j in Pz
276              if i!=j and (i,j) in t_all and t_all[(i,j)]<=
277                 MAX_TIEMPO_ARCO_PP],
278             key=lambda x:x[1]
279         )
280         for j,_ in vecinos[:K_VECINOS_PARADERO]:
281             A.add((i,j))
282
283     for j in Pz:
284         vecinos=sorted(
285             [(i,t_all[(i,j)]) for i in Pz
286              if i!=j and (i,j) in t_all and t_all[(i,j)]<=
287                 MAX_TIEMPO_ARCO_PP],
288             key=lambda x:x[1]
289         )
290         for i,_ in vecinos[:K_VECINOS_ENTRADA]:
291             A.add((i,j))
292
293     for i in Pz:
294         vecinos=sorted(
295             [(e,t_all[(i,e)]) for e in E if (i,e) in t_all],
296             key=lambda x:x[1]

```

```

295         )
296         for e,_ in vecinos[:K_VECINOS_ACCESO]:
297             A.add((i,e))
298
299     for e in E:
300         if (e,destino) in t_all:
301             A.add((e,destino))
302     return A
303
304 def verificar_conectividad(A,Pz):
305     O={}
306     I={}
307     for i,j in A:
308         O.setdefault(i,[]).append(j)
309         I.setdefault(j,[]).append(i)
310
311     alcanzados={origen}
312     q=deque([origen])
313     while q:
314         a=q.popleft()
315         for j in O.get(a,[]):
316             if j not in alcanzados:
317                 alcanzados.add(j)
318                 q.append(j)
319
320     llegan={destino}
321     q=deque([destino])
322     while q:
323         a=q.popleft()
324         for i in I.get(a,[]):
325             if i not in llegan:
326                 llegan.add(i)
327                 q.append(i)
328

```

```

329     sin_entrada=[i for i in Pz if i not in alcanzados]
330     sin_salida=[i for i in Pz if i not in llegan]
331     return sin_entrada,sin_salida
332
333 def construir_arcos_zona_factible(Pz):
334     A=construir_arcos_zona(Pz)
335     sin_entrada,sin_salida=verificar_conectividad(A,Pz)
336     if sin_entrada or sin_salida:
337         raise RuntimeError("Zona desconectada.")
338     return A
339
340 def obtener_solver():
341     for nombre_solver in ("appsi_highs","highs"):
342         s=pyo.SolverFactory(nombre_solver)
343         if s and s.available():
344             return s,nombre_solver
345         raise RuntimeError("Instalar HiGHS: pip install highs")
346
347 solver,solver_nombre=obtener_solver()
348
349 if solver_nombre=="appsi_highs":
350     solver.config.mip_gap=MIP_GAP
351     solver.config.time_limit=TIME_LIMIT_SEG
352 else:
353     solver.options["mip_rel_gap"]=MIP_GAP
354     solver.options["time_limit"]=TIME_LIMIT_SEG
355
356 def resolver_zona(Pz,kmax,zona):
357     A=list(construir_arcos_zona_factible(Pz))
358     t={a:t_all[a] for a in A}
359     s={i:(TIEMPO_DETENCION_PARADERO if i in Pz else 0) for i in N}
360     O={i:[j for a,j in A if a==i] for i in N}
361     I={i:[j for j,a in A if a==i] for i in N}
362     R=[f"Z{zona}-R{k}" for k in range(1,kmax+1)]

```

```

363
364 m=pyo.ConcreteModel()
365 m.A=pyo.Set(initialize=A,dimen=2)
366 m.R=pyo.Set(initialize=R)
367 m.x=pyo.Var(m.A,m.R,domain=pyo.Binary)
368 m.z=pyo.Var(m.R,domain=pyo.Binary)
369 m.subtour_cuts=pyo.ConstraintList()
370
371 def visita(mm,i,r):
372     return sum(mm.x[j,i,r] for j in I.get(i,[]) if (j,i) in A)
373
374 def tiempo_ruta(mm,r):
375     return (
376         sum(t[i,j]*mm.x[i,j,r] for i,j in A)
377         + sum(s[i]*visita(mm,i,r) for i in Pz)
378     )
379
380 m.cobertura=pyo.Constraint(
381     Pz,
382     rule=lambda mm,i: sum(visita(mm,i,r) for r in R)==1
383 )
384
385 m.salida=pyo.Constraint(
386     R,
387     rule=lambda mm,r: sum(
388         mm.x[origen,j,r]
389         for j in O.get(origen,[])
390         if (origen,j) in A
391     )==mm.z[r]
392 )
393
394 m.llegada=pyo.Constraint(
395     R,
396     rule=lambda mm,r: sum(

```

```

397         mm.x[i,destino,r]
398         for i in I.get(destino,[])
399             if (i,destino) in A
400                 )==mm.z[r]
401     )
402
403 m.continuidad=pyo.Constraint(
404     Pz+E,m.R,
405     rule=lambda mm,i,r:
406         sum(mm.x[j,i,r] for j in I.get(i,[]) if (j,i) in A)
407         ==
408         sum(mm.x[i,j,r] for j in O.get(i,[]) if (i,j) in A)
409 )
410
411 m.con_paradero=pyo.Constraint(
412     R,
413     rule=lambda mm,r: sum(visita(mm,i,r) for i in Pz)>=mm.z[r]
414 )
415
416 m.tmax=pyo.Constraint(
417     R,
418     rule=lambda mm,r: tiempo_ruta(mm,r)<=TMAX*mm.z[r]
419 )
420
421 m.simetria=pyo.Constraint(
422     range(len(R)-1),
423     rule=lambda mm,k: mm.z[R[k]]>=mm.z[R[k+1]]
424 )
425
426 m.obj=pyo.Objective(
427     expr=sum(tiempo_ruta(m,r) for r in R),
428     sense=pyo.minimize
429 )
430

```

```

431 def arcos_usados(mm,r):
432     return [
433         (i,j) for i,j in A
434         if pyo.value(mm.x[i,j,r]) is not None
435         and pyo.value(mm.x[i,j,r])>0.5
436     ]
437
438 def subtours(usados):
439     ruta=[origen]
440     a=origen
441     vistos={origen}
442     while a!=destino:
443         sig=[j for i,j in usados if i==a]
444         if not sig or sig[0] in vistos:
445             break
446         a=sig[0]
447         ruta.append(a)
448         vistos.add(a)
449
450     arcos_p=[(i,j) for i,j in usados if i in Pz and j in Pz]
451     pendientes={n for a in arcos_p for n in a}-set(ruta)
452     res=[]
453
454     while pendientes:
455         ini=next(iter(pendientes))
456         ciclo=[]
457         a=ini
458         vis={}
459         while True:
460             if a in vis:
461                 S=ciclo[vis[a]:]
462                 if len(S)>=2:
463                     res.append(set(S))
464                     break

```

```

465         if a not in pendientes:
466             break
467         vis[a]=len(ciclo)
468         ciclo.append(a)
469         sig=[j for i,j in arcos_p if i==a]
470         if not sig:
471             break
472         a=sig[0]
473         pendientes-=set(ciclo)
474     return res
475
476 for _ in range(MAX_ITER_CORTES):
477     res=solver.solve(m,tee=False)
478     term=str(res.solver.termination_condition).lower()
479     if term not in ["optimal","feasible"]:
480         raise RuntimeError("Modelo sin solucion factible.")
481
482     nuevos=0
483     for r in R:
484         if pyo.value(m.z[r]) is None or pyo.value(m.z[r])<0.5:
485             continue
486         for S in subtours(arcos_usados(m,r)):
487             AS=[(i,j) for i in S for j in S if (i,j) in A and i!=j]
488             if AS:
489                 m.subtour_cuts.add(
490                     sum(m.x[i,j,r] for i,j in AS)<=len(S)-1
491                 )
492                 nuevos+=1
493     if nuevos==0:
494         break
495
496 rutas=[]
497 for r in R:
498     if pyo.value(m.z[r]) is None or pyo.value(m.z[r])<0.5:

```

```

499         continue
500     usados=arcos_usados(m,r)
501     ruta=[origen]
502     a=origen
503     vistos={origen}
504
505     while a!=destino:
506         sig=[j for i,j in usados if i==a]
507         if not sig:
508             raise RuntimeError("No se pudo reconstruir la ruta.")
509         a=sig[0]
510         if a in vistos and a!=destino:
511             raise RuntimeError("Se detecto un ciclo.")
512         ruta.append(a)
513         vistos.add(a)
514
515     rutas.append((r,ruta))
516
517     return rutas,A,t
518
519 rutas_resultado=list(rutas_fijas_resultado)
520 t={}
521
522 for _,ruta in rutas_fijas_resultado:
523     for i,j in zip(ruta[:-1],ruta[1:]):
524         t[(i,j)]=t_all[(i,j)]
525
526 for z,Pz in zonas.items():
527     kmax=KMAX_SIN_ZONAS
528     rz,Az,tz=resolver_zona(Pz,kmax,z)
529     rutas_resultado.extend(rz)
530     t.update(tz)
531
532 resumen_rutas=[]

```

```

533 detalle_rutas=[]
534
535 for r,ruta in rutas_resultado:
536     tiempo_viaje=sum(t[(i,j)] for i,j in zip(ruta[:-1],ruta[1:]))
537     tiempo_detencion=sum(
538         TIEMPO_DETENCION_PARADERO
539         for n in ruta
540         if tipo_nodo[n]=="PARADERO"
541     )
542     distancia=sum(
543         distancia_all[(i,j)]
544         for i,j in zip(ruta[:-1],ruta[1:])
545     )
546
547     resumen_rutas.append({
548         "ruta":r,
549         "secuencia":[nombre[n] for n in ruta],
550         "paraderos":sum(tipo_nodo[n]=="PARADERO" for n in ruta),
551         "distancia_km":round(distancia,2),
552         "tiempo_total_min":round(tiempo_viaje+tiempo_detencion,2)
553     })
554
555     tiempo_acum=0
556     distancia_acum=0
557
558     for orden,nodo in enumerate(ruta,1):
559         if orden>1:
560             anterior=ruta[orden-2]
561             distancia_acum+=distancia_all[(anterior,nodo)]
562             tiempo_acum+=t[(anterior,nodo)]
563
564         if tipo_nodo[nodo]=="PARADERO":
565             tiempo_acum+=TIEMPO_DETENCION_PARADERO
566

```

```
567     detalle_rutas.append({
568         "ruta":r,
569         "orden":orden,
570         "nombre":nombre[nodo],
571         "tipo":tipo_nodo[nodo],
572         "distancia_acum_km":round(distancia_acum,2),
573         "tiempo_acum_min":round(tiempo_acum,2)
574     })
575
576 df_resumen=pd.DataFrame(resumen_rutas)
577 df_detalle=pd.DataFrame(detalle_rutas)
```