



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

Implementación de un Sistema de Planificación de Rutas Óptimas Multimodales en el
Transporte Público de Santiago

PROPUESTA DE TEMA DE MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

Javier Andrés Facondi Moran

MODALIDAD:
Memoria

PROFESOR GUÍA:
Eduardo Graells G.

SANTIAGO DE CHILE
2025

1. Introducción

El proyecto Ayatori [1] tuvo como objetivo principal la creación de un módulo en Python que sirve de base para el desarrollo de algoritmos de movilidad y generación de rutas, utilizando la información cartográfica de Santiago (proveniente de OpenStreetMap - OSM) y los datos de la Red Metropolitana de Movilidad (en formato GTFS). Para validar su utilidad, Ayatori implementó una versión simplificada de Connection Scan Algorithm (CSA), un algoritmo diseñado para responder eficientemente a consultas en sistemas de información de horarios y que optimiza los tiempos de viaje.

Sin embargo, esta primera implementación del CSA fue deliberadamente simplificada, omitiendo la funcionalidad de calcular transbordos entre distintos recorridos del transporte público (bus-bus, bus-Metro, etc.). Esta omisión limita severamente la capacidad de Ayatori para modelar la movilidad real de la ciudad, ya que los habitantes de Santiago utilizan los servicios del Metro de Santiago, los buses Red, y el tren urbano Nos-Estación Central en múltiples combinaciones, generando un elevado número de rutas diferentes. Esta restricción convierte al módulo en una herramienta de demostración con una utilidad limitada para un usuario final que necesita moverse por la ciudad.

La presente propuesta se centra en superar esta limitación algorítmica mediante la implementación de un algoritmo de planificación de rutas robusto (como la versión completa de CSA) que incorpore de manera eficiente la lógica de transbordos multimodales y los segmentos de caminata necesarios para conectar los servicios. Esto permitirá, por ejemplo, calcular rutas complejas como “¿Cómo llego desde Beauchef 851 hasta la Biblioteca Municipal de Pudahuel?” optimizando no solo el tiempo de espera y el tiempo de viaje, sino también la cantidad de transbordos. Por “multimodal” entendemos la integración de diferentes servicios de transporte (buses, Metro, tren urbano) y los segmentos de caminata que los unen.

Este desarrollo es crucial, ya que transformará al módulo Ayatori en un sistema completo de enrutamiento, contribuyendo con datos concretos a las organizaciones encargadas de administrar los servicios de transporte para la evaluación de la eficiencia de rutas existentes o la planificación de nuevas conexiones. Esto sentará las bases para futuras investigaciones de movilidad urbana.

2. Situación Actual

El módulo Ayatori almacena y opera con grandes volúmenes de datos utilizando grafos gestionados por la librería `graph-tool`, elegida por su eficiencia superior a alternativas como `networkx`. Los datos se organizan en dos clases principales: `OSMGraph` (para la infraestructura cartográfica urbana) y `GTFSData` (para la red de transporte y cronogramas).

A pesar de su base técnica sólida, la versión simplificada de CSA (*connection scan lite*) implementada en Ayatori presenta varias falencias que deben ser abordadas:

1. **Omisión de Transbordos:** La principal limitación es que la solución actual solo encuentra rutas directas entre paradas, delegando la complejidad del cálculo de transbordos (cambios de servicio) a posibles trabajos futuros [2].
2. **Dependencia de Datos Estáticos:** La implementación utiliza información estática basada en cronogramas, por lo que opera bajo el supuesto de una continuidad operativa constante. Esto puede generar un sesgo importante en horarios complejos como las horas punta, ya que no considera variables reales como embotellamientos o retrasos [5].
3. **Responsabilidad Manual del Usuario:** El usuario es responsable de descargar y mantener la última versión del archivo GTFS (`gtfs.zip`) para evitar sesgos por desactualización, lo que limita la usabilidad. La gestión de datos GTFS es un problema conocido en el desarrollo de este tipo de sistemas [8].

3. Objetivos

Objetivo General

Desarrollar e implementar la versión completa del Connection Scan Algorithm (CSA) o un algoritmo de planificación de rutas óptimas equivalente en Python, extendiendo el módulo Ayatori para soportar transbordos multimodales y segmentos de caminata, con el fin de generar rutas eficientes y realistas en el transporte público de Santiago, y así contribuir al estudio avanzado de la movilidad urbana.

Objetivos Específicos

1. Investigar y analizar las estructuras de datos y metodologías del Connection Scan Algorithm que permitan modelar y optimizar transbordos (múltiples cambios de servicio) y segmentos peatonales entre servicios [2].
2. Diagnosticar las limitaciones del módulo Ayatori y extender su estructura de datos (`GTFSData`) para modelar eficientemente la conectividad multimodal, integrando el cálculo de distancias para transbordos a pie mediante la fórmula Haversine [3].
3. Implementar el algoritmo de planificación de rutas óptimas (CSA completo o similar) que encuentre la ruta con la secuencia de servicios y transbordos más eficiente, considerando tiempo de espera, tiempo de viaje y tiempo de caminata [7].
4. Generar visualizaciones detalladas y georreferenciadas de las rutas multimodales encontradas, utilizando las funcionalidades cruzadas de OSM y GTFS (como el método `find_route_nodes` y la librería `folium`), destacando los segmentos de tránsito y los segmentos de caminata [6].

Evaluación

Se evaluará la eficiencia y funcionalidad del sistema mediante la ejecución de casos de prueba reales de enrutamiento en Santiago, validando las rutas generadas contra soluciones comerciales existentes.

4. Solución Propuesta

La solución se implementará directamente sobre la base de código del módulo Ayatori, centrándose en el desarrollo de una nueva lógica de enrutamiento que sustituya o complemente a *connection scan lite*.

El desafío principal es la gestión de los transbordos. La solución requerirá:

1. **Construcción de un Supergrafo Implícito:** Utilizar los grafos existentes por ruta de `GTFSData` y la información de `OSMGraph` para tratar las paradas (*stop id*) y estaciones como nodos de transferencia. Esto permitirá calcular las distancias de caminata (*haversine*) entre paradas cercanas que pertenecen a diferentes servicios, modelando así el transbordo como un segmento de caminata con un costo de tiempo asociado [8].
2. **Optimización Multicriterio (CSA):** Implementar la lógica de CSA para generar soluciones Pareto-eficientes, optimizando simultáneamente el tiempo total de viaje y el número de transbordos. El algoritmo CSA es particularmente adecuado para este problema porque su naturaleza de iteración cronológica facilita la consideración de tiempos de espera y transbordos de forma eficiente [2].
3. **Mejora de Usabilidad y Robustez:** Si bien el objetivo principal es la implementación del algoritmo, se explorará la posibilidad de integrar métodos para la actualización automática de datos de GTFS (mencionado como un trabajo futuro en la memoria anterior) o, al menos, fortalecer la robustez del sistema ante la falta de datos en tiempo real.

5. Plan de Trabajo (Preliminar)

5.1. Semestre Actual 2025-2 (Semanas 7-15): Fase de Preparación y Diseño de Extensión

Este semestre se dedica a la investigación teórica, el diagnóstico de las falencias del módulo Ayatori (más allá de la omisión de transbordos) y la adecuación del código base para soportar la implementación algorítmica mayor en el semestre de finalización.

1. Análisis y Adquisición de Conocimiento (Semanas 7-9)

- (a) Revisión bibliográfica exhaustiva del Connection Scan Algorithm (CSA) y sus variantes que permiten la optimización de rutas considerando transbordos (cambios de servicio) [2].
- (b) Identificación de las falencias y limitaciones del módulo base Ayatori, incluyendo la dependencia de datos estáticos y el uso de versiones de GTFS que deben ser actualizadas manualmente por el usuario [8, 5].
- (c) Estudio detallado de las estructuras de datos necesarias para modelar eficientemente los transbordos multimodales, definiendo cómo se utilizarán las coordenadas de las paradas (contenidas en `route_stops`) y la fórmula *haversine* para calcular distancias de caminata entre paradas [3].

2. Adecuación y Diseño del Modelo (Semanas 9-13)

- (a) Asegurar la funcionalidad completa y estabilidad del módulo base Ayatori (procesamiento `pyrosm` y `pygtfs`), corrigiendo cualquier fallo en las funciones existentes de grafos (`graph-tool`) o manejo de datos.
 - (b) Diseño de la arquitectura de datos para la gestión de transbordos, extendiendo la clase `GTFSData` para manejar la conectividad lógica entre servicios que no tienen conexión directa.
 - (c) Prototipado y validación de las funciones auxiliares de cálculo de tiempo de caminata entre paradas cercanas, asumiendo una velocidad de caminata fija [9].
- 3. Documentación (Semanas 14-15)**
- (a) Redacción del Avance de Memoria y documentación técnica del diseño extendido, incluyendo el marco teórico de CSA completo y la preparación del código para la implementación algorítmica.

5.2. Semestre 2026-1 (Semanas 1-15): Implementación, Evaluación y Redacción Final

Este semestre se enfoca en la implementación del algoritmo principal, la validación funcional, el desarrollo de visualizaciones y la finalización de la documentación.

- 4. Implementación Algorítmica (Semanas 1-9)**
- (a) Desarrollo e integración del Connection Scan Algorithm (CSA) completo en Python, capaz de optimizar la ruta de forma Pareto-eficiente, minimizando el tiempo de viaje total y el número de transbordos [4, 2].
 - (b) Implementación de la lógica de enrutamiento multimodal, incorporando el cálculo de segmentos de caminata y la búsqueda eficiente de puntos de transbordo (paradas y estaciones de Metro) [7].
- 5. Pruebas de Funcionamiento y Visualización (Semanas 9-12)**
- (a) Implementación de visualizaciones georreferenciadas (utilizando `folium`) de las rutas encontradas, diferenciando claramente los segmentos recorridos en transporte público y los segmentos de caminata entre transbordos y puntos finales [6].
 - (b) Ejecución de casos de estudio reales de enrutamiento en Santiago (Ej: Beauchef 851 a Biblioteca Municipal de Pudahuel) y pruebas de validación funcional, comparando los resultados del CSA completo con las rutas generadas por soluciones comerciales [6].
- 6. Redacción y Conclusión Final (Semanas 12-15)**
- (a) Análisis y consolidación de los resultados obtenidos, incluyendo la medición de la eficiencia y la calidad de las rutas generadas por el algoritmo.
 - (b) Redacción y consolidación de la memoria final (extendiendo las semanas de trabajo dedicadas a este punto), incluyendo la discusión de las implicancias y limitaciones del sistema (como la dependencia en datos estáticos de GTFS) [5].
 - (c) Preparación de la defensa de la memoria, incluyendo la generación de material de apoyo visual que demuestre el funcionamiento del sistema con transbordos. (Cierre del Objetivo General).



Figura 1: Carta Gantt Semestre 2026-1

Referencias

- [1] Cerro, Felipe Ignacio Leal: *Ayatori: Creación de módulo base para programar algoritmos de planificación de rutas Python usando GTFS*. Tesis de Doctorado, Universidad de Chile, 2023. <https://repositorio.uchile.cl/handle/2250/198845>, Memoria para optar al título de Ingeniero Civil en Computación.
- [2] Dibbelt, Julian, Thomas Pajor, Ben Strasser y Dorothea Wagner: *Connection Scan Algorithm*. CoRR, abs/1703.05997, 2017. <http://arxiv.org/abs/1703.05997>.
- [3] Hertell, Daniel y Jari J. Korhonen: *Multimodal Pathfinding with A-star and GTFS Data*. En *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*, páginas 407–412. ACM, 2016.
- [4] Müller, Jan Peter y Christian P. T. Zündorf: *A Survey of Algorithms for Shortest Paths in Timetable Networks*. Journal of Experimental Algorithmics, 18(1):1.1–1.31, 2013.
- [5] Müller-Hannemann, Matthias y Mathias P. T. Zündorf: *Route Planning in Public Transportation Networks*. Encyclopedia of Algorithms, páginas 709–713, 2007.
- [6] Nogueira, Guilherme, Bruno H. C. Silva y Ana Carolina M. da Silva: *Using Open Data for Urban Mobility Analysis and Visualization*. Applied Sciences, 10(15):5205, 2020.
- [7] Ordoñez, Francisco, Carolina Salinas, Luis F. Escobar y Claudia A. García: *Multimodal Path Planning with Walking and Public Transit*. Computers & Industrial Engineering, 113:130–140, 2017.
- [8] Ortiz-Rosales, M., J. L. Tiznado-Astudillo y D. F. Pizarro-Torres: *A Multimodal Routing Algorithm for Public Transportation using GTFS data*. Procedia Computer Science, 153:114–121, 2019.
- [9] Zografos, Konstantinos y Kostas Androustopoulos: *A new algorithm for the single-vehicle shortest path problem in multimodal transportation networks*. Transportation Research Part B: Methodological, 41(10):1114–1130, 2007.