UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

MODELAMIENTO DE INCERTIDUMBRE EN LOS TIEMPOS DE VIAJE

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN GESTIÓN DE OPERACIONES

FELIPE ANDRÉS LAGOS GONZÁLEZ

PROFESOR GUÍA:
FERNANDO ORDOÑEZ PIZARRO

MIEMBROS DE LA COMISIÓN:
DANIEL ESPINOZA GONZÁLEZ
MARCELA MUNIZAGA MUÑOZ

SANTIAGO DE CHILE
2014

Resumen

Cada día Santiago de Chile se convierte en una ciudad con más habitantes y, consecuentemente, con un mayor número de vehículos. El Cuerpo de Bomberos de Santiago (CBS) se le presenta, entonces, un gran desafío, pues debe atender a una emergencia en poco tiempo y al mismo tiempo lidiar con calles y avenidas congestionadas.

El Departamento de Ingeniería Industrial de la Universidad de Chile (DII), ha desarrollado herramientas que apuntan a encontrar rutas óptimas para llegar a una emergencia con el fin de ayudar en su labor a CBS. Sin embargo, estas aplicaciones, hasta el momento, no utilizan criterios que incluyan la distribución de probabilidad de estos tiempos, lo que podría ayudar a tomar mejores decisiones.

Usando datos del sistema de transporte urbano de Santiago, Transantiago, se busca estudiar la distribución de los tiempos de viaje por arcos de un grafo que representa esta ciudad. Se propone una metodología que abarca desde el manejo de estos datos, hasta un modelo que permite estimar distribuciones.

Inicialmente, se sugiere un modelo de datos con sus índices. Luego, se da paso a la descripción de métodos para la identificación de rutas seguidas por los distintos servicios. Haciendo uso de algoritmos de proyección y Cadenas de Markov, se logran procesar más de 5300 arcos, los que posteriormente, se utilizan para proyectar tiempos de viaje y velocidades. Con los datos procesados, estas variables aleatorias se estudian en su distribución de probabilidad, comportamiento a lo largo de un camino y criterios de ajuste.

Los tiempos de viaje muestran ser una variable aleatoria de distribución Lognormal para una gran cantidad de arcos. Los resultados obtenidos, además, permiten encontrar un perfil para los arcos que presentan un mejor ajuste, tanto por sus características espaciales como temporales. Junto con ello, se resuelve si estos tiempos son independientes o no. Finalmente, en base a los resultados se establece que es importante incluir la correlación entre los tiempos de viaje.

Se estudia la suma de tiempos en caminos arbitrarios, comparándola con datos reales. Los resultados permiten validar el modelo propuesto, e identificar qué método para la suma de variables es mejor. Finalmente se analizan distribuciones para bloques de horarios, concluyendo que los tiempos de viaje es mejor tratarlos de la forma más desagregada posible.

Para trabajos futuros se recomienda analizar mezclas de Lognormales para los tiempos de viaje y estudiar la distribución Burr como una alternativa.

Agradecimientos

Agradezco todo el apoyo que me ha dado mi familia a lo largo de estos años, han sido un pilar muy importante para que todo esto haya sido posible. También quiero agradecer a mi novia, María Francisca, quien me ha acompañado todos meses de trabajo.

Finalmente dar gracias a mis profesores de tesis, Fernando y Daniel, por la ayuda a lo largo de todo el proyecto. Guiaron mi trabajo y semana a semana me dieron comentarios y sugerencias de cómo abordar los distintos problemas a los que me enfrentaba.

Tabla de contenido

Índice de tablas

Índice de figuras

Introducción

Santiago es la mayor ciudad de Chile y una de las más importantes de Sudamérica. Para el último censo con resultados publicados, se determinó que la ciudad era el hogar de más de 5.6 millones de habitantes, lo cual correspondía a aproximadamente un 37 % del total de personas del país [21]. Consecuentemente, alberga las principales instituciones administrativas, culturales, financieras, etc. de Chile.

Tiene una extensión de más de 600 km^2, lo que la deja con una densidad promedio de 78 habitantes por hectárea. Este número es bajo para estándares de países desarrollados, sin embargo, existen comunas donde la tasa asciende a 150 hab./ha. Junto con el crecimiento y condiciones demográficas van las económicas, gracias a las cuales, Santiago se ha convertido en una de las ciudades más competitivas de latinoamérica y del mundo[1]. Este crecimiento, naturalmente, también ha alcanzado a la industria automotriz, expresado en la cantidad de autos que actualmente circulan por las calles de Santiago. En el año 2004, se estima que habían casi 16.5 millones de viajes durante un día de trabajo típico, esto equivalía a 3 viajes por habitante en promedio al día. Desafortunadamente, la capacidad de vial de la ciudad no ha crecido a la par con el parque automotriz, traduciéndose en atochamientos o embotellamientos por doquier. Hasta 2 horas puede tardar una persona en llegar a su hogar en la tarde [21].

Gran parte de las calles, avenidas, autopistas, etc. están saturadas en horas punta, e incluso llegan a estarlo en horarios regulares. Es clave, entonces, definir métodos y sistemas que permitan eficiente encontrar rutas para desplazarse desde un punto de la ciudad a otro, y aprovechar así, la capacidad disponible lo mejor posible. Este desafío es aún mayor para quienes deben manejar emergencias, como por ejemplo, el cuerpo de bomberos.

Cada día se reciben muchas llamadas a las distintas compañías que tiene el cuerpo de bomberos en la ciudad. La misión de cada una de ellas es llegar a la emergencia informada, en el menor tiempo posible, y actuar rápidamente en la forma que se requiera. La decisión de qué camino escoger, sin embargo, hasta hace no mucho tiempo, se hacía en la central de despacho de forma predeterminada y estática. No se calculaba explícitamente estimaciones de los tiempos de viaje a la emergencia para definir las prioridades en los despachos. Dentro del proyecto global en que este estudio se enmarca, uno de los avances que ya ha sido implementado corresponde a un algoritmo de caminos mínimos para obtener prioridad de despacho considerando las mejores rutas en un grafo de Santiago a distintas horas del día. El criterio que se usa es minimizar el tiempo promedio total de traslado. No obstante, el tiempo

[1] http://www.demographia.com/db-world-metro2000.htm. *50 largest world metropolitan areas ranked*

de viaje ha demostrado ser una variable con un comportamiento aleatorio importante, lo que hasta ahora no ha sido incorporado.

En particular, para un despacho de vehículos en el evento de una emergencia considerar la incertidumbre en los tiempos de viaje es importante. Si bien pueden existir rutas que tienen el mismo tiempo promedio desde un mismo inicio y destino, el riesgo que tenga una u otra pueden marcar la diferencia.

¿Cómo se puede representar este tiempo? ¿Cuál es su naturaleza? ¿Qué tan distinto es en un camino con respecto a otro? ¿Cambia a lo largo del día este tiempo de viaje? son solo algunas preguntas que se pueden formular y que se buscará responder usando información disponible de quienes pasan más tiempo en las calles de Santiago: el sistema de transporte público Transantiago. Transantiago es el nombre que recibe el sistema completo de buses públicos que diariamente siguen las distintas rutas que se han establecido para Santiago. Esta red de vehículos recorre gran parte de la ciudad durante todo el año.

En este trabajo de tesis se busca construir y analizar distribuciones de los tiempos de viaje inciertos que han registrado los buses del Transantiago, y con ello, dar paso posteriormente a modelos de ruteo que sean capaces de incluir diversos factores de incertidumbre. Dos son los principales puntos a tomar en cuenta como aportes:

- Se analizan las distribuciones que siguen los tiempos de viaje para distintos arcos de una parte del grafo de Santiago
- Se estudia la mejor forma de relacionar los tiempos de viaje de forma de estimar distribuciones por caminos arbitrarios, considerando posibles correlaciones entre estas variables aleatorias.

Capítulo 1

Contexto general

Esta tesis de magister se llevó a cabo como parte del proyecto *D10I1002: Tecnología avanzada para ciudades del futuro*, financiado por CONICYT a través del Fondo de Fomento al Desarrollo Científico y Tecnológico (FONDEF).

Cada uno de los proyectos que son asignados por este fondo están descritos en la página de esta institución[1], donde, además, es posible encontrar quienes son los responsables, objetivos del proyecto, financiamiento, categoría, etc., todo esto con el fin de hacer el proceso transparente a la comunidad. En particular, para el proyecto marco de este estudio se publica:

> **Problema:** Se requiere una adecuada planificación de las ciudades en términos de su expansión, servicios de emergencia y transporte.
> **Solución:** Conjunto de software de apoyo a la gestión y toma de decisiones. Específicamente: Gestión de emergencias atendidas por bomberos (determina la estación de bomberos que debe concurrir a un llamado y planifica la localización de estaciones); explotación de datos de transporte público (por ejemplo: indicadores de desempeño, tiempos de espera y viaje, cumplimiento de frecuencias, etc.). La planificación urbana proyecta la evolución de la ciudad bajo distintos escenarios (proyectos de transporte, cambios de planes reguladores, etc.) y determina el paquete óptimo de medidas.
> **Institución:** U de Chile. Mandante: Subsecretaría de Transporte del MTT. Asociada: Subsecretaría de Vivienda del MINVU; Ministerio de Vivienda y Urbanismo; Cuerpo de Bomberos de Santiago.

Como bien se describe, el proyecto tiene tres aristas o enfoques que se va a abordar. Uno de estos es la gestión de emergencias atendidas por el Cuerpo de Bomberos de Santiago (CBS), entendido este punto como la construcción e implementación de herramientas que permitan dar soporte a la toma de decisiones de ruteo y gestión de vehículos. El CBS es responsable de atender nueve comunas de la capital, lo cual es solo una parte de la ciudad. El resto de las comunas están bajo la dirección de otros Cuerpos de Bomberos. A pesar de ello, el CBS es el cuerpo de bomberos más grande de la región metropolitana, cubriendo anualmente más de 5 mil llamadas de emergencias.

[1] http://www.fondef.cl

Como toda organización o institución que tiene que acudir rápidamente a una emergencia, el objetivo principal que se desea alcanzar con CBS es el de despachar los vehículos de emergencia de manera de llegar a la urgencia en el menor tiempo posible. Unos pocos minutos de demora pueden significar grandes daños materiales o incluso la vida de víctimas. Debido a ello, la solución que se está elaborando busca integrar métodos y algoritmos que ayuden a tomar decisiones óptimas. Específicamente se busca evaluar y diseñar un sistema de despacho basado en criterios netamente de optimización que incorpore elementos de velocidad a distintas horas del día en las calles y reversibilidad de las mismas [9].

El CBS está constituido por 22 compañías o centros distintos desde los cuales se acude a la emergencia. Estos centros están distribuidos principalmente en la zona centro - oriente de la ciudad concentrados principalmente en el centro [9]. Actualmente, cada una de estas compañías está participando activamente del proyecto. En la Figura 1.1 está la zona controlada únicamente por el CBS. Las 9 compañías al alero del CBS tienen que atender emergencias comprendidas en esta zona, y deben respectar las zonas de otros Cuerpos de Bomberos. Es por ello que las rutas que debe sugerir el sistema de despacho tienen el impedimento de que deben estar contenidas, en la medida de lo posible, en esta zona.

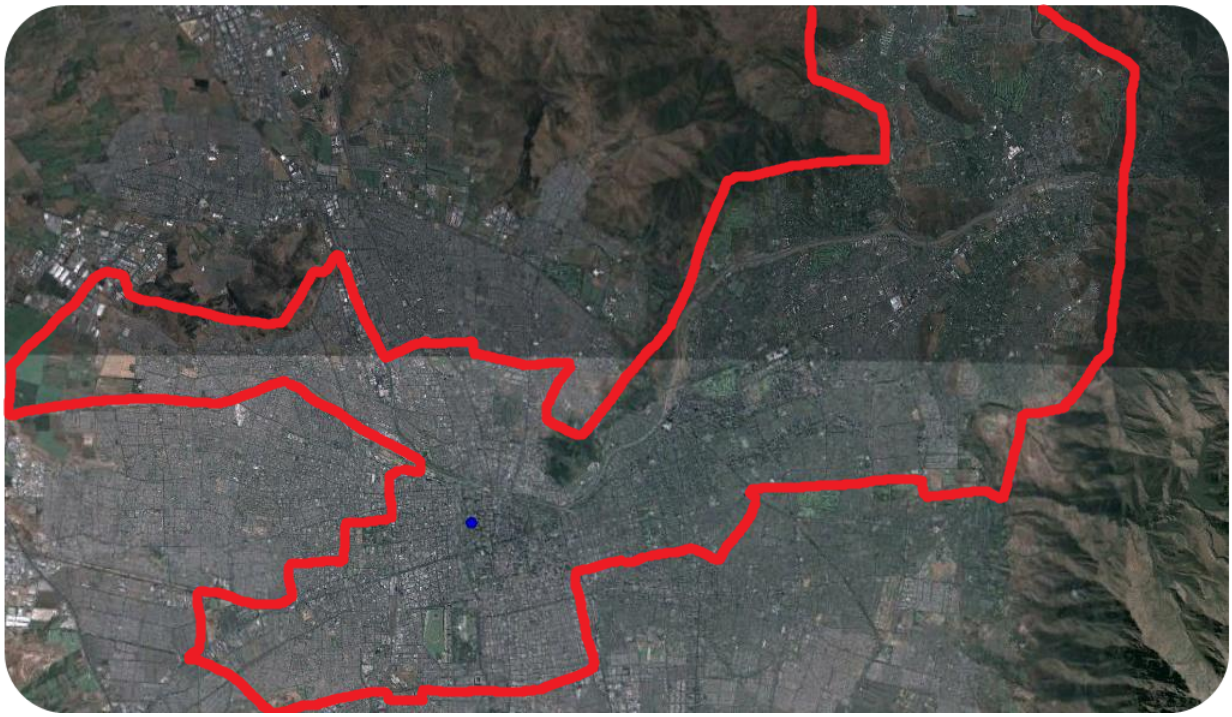


Figura 1.1: Zona de despacho de las compañías asociadas al proyecto

El CBS está llevando a cabo una serie de mejoras en su sistema de despacho, para lo cual establecido metas para mejorar su sistema de atención de emergencias. Algunas de estos objetivos son alcanzados con la ayuda del proyecto global en el que se enmarca este estudio:

- Definir el número de vehículos para cada emergencia. Además se debe considerar que dependiendo del tipo de siniestro es el tipo de máquinas requerido.
- Una vez evaluada la situación en terreno es posible levantar nuevas alarmas. Determinar dónde se dará la alarma.

- No siempre las máquinas están disponibles. Incorporar este factor en las decisiones.
- Todo el sistema de despacho es controlado desde una central de despacho, por lo tanto son ellos quienes deben manejar la interfaz propuesta.
- La central, una vez ingresada la dirección de la emergencia, debe recibir como respuesta el orden con el cual las máquinas deben ser despachadas.
- Desarrollar un sistema de despacho que incluya mapa de calles y sentido, congestión en distintas horas del día, que sea capaz de entregar un orden según el tiempo de respuesta de cada compañía para una emergencia dada.

El problema de despacho se plantea como uno de caminos mínimos, es decir, dado un grafo encontrar la ruta que minimiza el costo de llegar de un punto a otro. El grafo se representa como un conjunto de nodos y arcos (con sentido) que unen estos nodos. El costo de atravesar uno de estos arcos está determinado y es fijo, y el objetivo es establecer el camino entre un punto y otro que minimice el costo total. Este problema no es nuevo, y ya desde hace varios años que se estudia. Uno de los primeros autores en desarrollar algoritmos para resolver problemas de esta naturaleza fue *Dijkstra* [8], precursor de una linea de investigación de varias décadas. Su algoritmo es uno de los más conocidos para caminos mínimos, a pesar de que cuenta con la restricción de que los costos deben ser estrictamente positivos. Afortunadamente, el problema de ruteo en tiempo mínimo no viola este supuesto, puesto que los tiempos siempre son mayores que cero, permitiendo así, utilizar el algoritmo de *Dijkstra* para resolverlo. Con esto se tiene garantía de encontrar el camino óptimo en tiempos polinomiales [3].

El grafo de Santiago sobre el cual se rutea fue construido a partir de una cartografía de la ciudad por Publiguías. Esta empresa donó a CBS el mapa para dar soporte a los despachos, tiempo antes de comenzar el proyecto [9]. Aquí se tienen los ejes viales actualizados hasta el año 2008, además de puntos de interés y ubicación Global Positioning System (GPS) de las compañías afiliadas al proyecto. La información se encuentra almacenada en archivos de texto plano (.txt), dentro de los cuales está un archivo para los nodos y otro para los arcos. Además, es posible identificar información adicional, como por ejemplo, tipo de ejes viales. Se representan todas las calles, avenidas, carreteras, etc. de Santiago, y dentro del detalle se tiene:

- Ubicación de los extremos de cada arco (nodos del grafo)
- Sentido de tránsito (arcos con sentido)
- Numeración correspondiente al segmento
- Nombre de la calle, avenida, etc
- Comuna dónde se ubica

En una primera fase de este proyecto, se buscó identificar los tiempos de viaje por las distintas vías de Santiago, haciendo uso de datos procesados de velocidad. El objetivo que se planteó fue incorporar datos de velocidad y tiempo de viaje al sistema que obtiene el camino mínimo. En otras palabras, había que incorporar un costo a los arcos que reflejara de la mejor forma el tiempo real para transitar por los distintos ejes. El origen de estos primeros datos corresponde al sistema de transporte público de Santiago, Transantiago, los que en forma agregada reportaban la velocidad para segmentos rectos de 500 metros. Estas polilíneas no

guardaban directa relación con el grafo de Santiago, y, a pesar de que estas rectas estaban identificadas con latitud y longitud, los caminos no eran fáciles de relacionar.

La información suministrada, además de contar con estas velocidades, reportaba el número de buses que se utilizaron para construir esa velocidad, lo que permitió estimar errores para diferentes momentos de las variables aleatorias. Los buses seguían rutas definidas las cuales estaban compuestas por polilíneas, correspondientes a diferentes servicios.

Consecuentemente, el primer paso para estimar tiempos de viaje para el ruteo de vehículos fue identificar por cuáles arcos del grafo circulan los recorridos de buses. El grafo es mucho más denso que lo que estaba incorporado en los datos, ya que en 500 m. pueden fácilmente hallarse 10 arcos. Una vez seleccionados estos arcos, se les asoció la velocidad de los buses registrada en el segmento correspondiente, eligiendo, a su vez, la mejor forma de distribuir dicha información.

Usando heurísticas de proyección se consiguió detectar el camino que seguía cada servicio, en el mapa. Estos algoritmos, en un tiempo breve, proyectaban los puntos extremos de cada polilínea en el grafo sobre los arcos más cercanos. Luego, dentro de los arcos candidatos buscaba la ruta seguida por el vehículo. A pesar de que el tiempo necesario para identificar estos arcos es menor al que hubiese tomado un algoritmo óptimo, presentaba problemas para ciertos tipos de recorridos, los cuales, en muchas ocasiones, no era posible asociar la ruta correcta [9].

Posteriormente, sobre estas rutas se proyectaron las velocidades. Sin embargo, no todas las vías de Santiago son cubiertas por algún bus del trasporte público, por lo cual, fue necesario extrapolar algunos resultados. De esta manera, se definieron perfiles de velocidad por defecto para cada uno de los tipos de calles en el grafo con variaciones según el horario del día.

Con las velocidades proyectadas y otras establecidas por defecto, se pudo calcular el tiempo de traslado, con lo cual fue posible implementar una primera versión para el despacho de vehículos de emergencia. El algoritmo utilizado para encontrar la mejor ruta es una versión de Dijkstra que implementa un heap binario que simplifica la búsqueda [9]. Un heap binario es, a grandes rasgos, una estructura de datos de árbol binario balanceado, la cual impone algunas restricciones extras con el fin de evitar almacenar el grafo de Santiago completo. Esta estructura de datos mejora considerablemente el tiempo de resolución del algoritmo, puesto que las búsquedas en conjuntos menores [15].

Adicionalmente, se incorporó una segunda mejora, la que evitó tener que resolver el algoritmo para cada par compañía-emergencia. Debido a que basta con resolver el camino más corto desde todos los orígenes a un único destino, se ejecuta el algoritmo tomando como nodo de partida la emergencia y como destino cada uno de los centros de CBS. Esta alternativa se conoce como Dijkstra reverso.

Sin embargo, como se mencionó anteriormente, Dijkstra asume costos fijos positivos para todos los arcos del grafo. Debido a que lo que se busca minimizar es el tiempo total de traslado, el supuesto de que los tiempos sean positivos no tiene ningún problema. El inconveniente se presenta cuando se tiene que representar estos tiempos como valores fijos. El supuesto de que los tiempos por un arco van a ser siempre los mismos, es muy fuerte y no es un muy buen

reflejo de la congestión que se ve en la realidad.

Una primera aproximación consistió en dividir el día en segmentos de 30 minutos, cada uno de ellos con un perfil de viaje distinto. Se crean 48 diferentes momentos del día para incorporar la velocidad y tiempo de viaje de un arco. Gracias a esto, se consigue representar uno de los horarios más rápidos para trasladarse, como el intervalo entre las 6:00 y 6:59 hrs., con un patrón de viaje distinto al que pueda tener el de las 18:00 y 18:59 hrs., uno de los más congestionados. Usando las velocidades proyectadas y las velocidades por defecto, se estimaron perfiles de viaje para cada uno de los arcos del grafo, agregando así los costos del problema.

El CBS comenzó a usar este sistema para decidir los despachos desde diciembre del 2012. Uno de las herramientas que se entregó es la que permite dar prioridad a los despachos desde diferentes centros. Este programa está desarrollado en el lenguaje *C*, el cual además permite soportar una interfaz desarrollada usando *PHP, HTML* y *JS*. La herramienta web está recogida en la Figura 1.2. Aquí es posible marcar una emergencia (con coordenadas dadas por los campos latitud y longitud de la interfaz) y obtener las rutas que deben seguirse desde los diferentes centros del CBS. Las prioridades también son desplegadas en la página en el cuadro de la izquierda.

A la fecha la institución se ha mostrado muy contenta con este nuevo sistema de despacho, en gran parte, debido a que constantemente CBS ha estado participando en la implementación y desarrollo de esta herramienta.

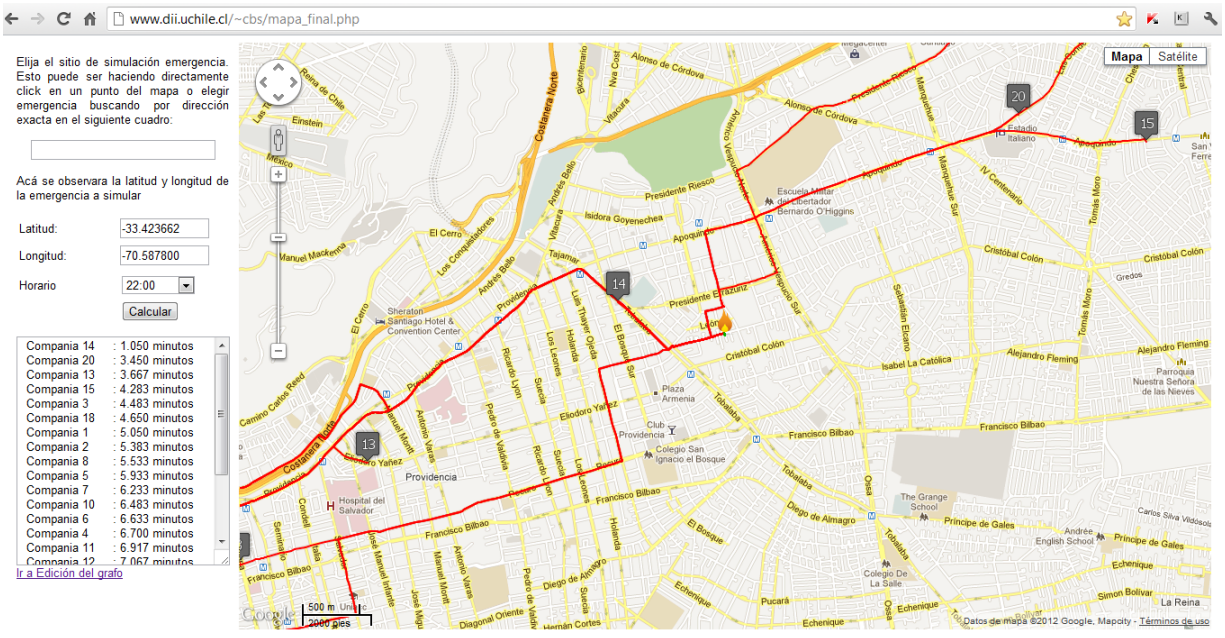


Figura 1.2: Interfaz utilizada por la Central de despacho de CBS

No obstante, la información y los criterios que se usan para rutear las emergencias, salvo por la incorporación de perfiles horarios, permanecen sin incertidumbre. En su defecto, se considera el tiempo de viaje esperado como medida de costo, lo cual no siempre es un buen criterio para decidir. Un ejemplo que muestra que ignorar la distribución de probabilidad es problemático se puede encontrar en Finanzas. El problema clásico en esta materia es la

composición de una cartera de activos, los cuales tienen retornos aleatorios en el tiempo. *Markowitz* [17] propuso un modelo para la conformación de un portafolio de valores con distintos niveles de rentabilidad. Su modelo toma en cuenta tanto el retorno (valor esperado del portafolio) como el riesgo (varianza), lo que permite establecer un equilibrio entre un componente y otro, reflejando que el criterio del valor esperado de rentabilidad no es el único que se toma en cuenta. Si solamente se buscara maximizar el valor esperado, entonces el portafolio óptimo sería aquel que invierte todo en el activo más rentable, lo cual no sería recomendable puesto que tendría un riesgo muy alto.

Cuando lo que se busca es encontrar la mejor ruta para llegar a una emergencia, el tiempo esperado del camino es normalmente el criterio a tener en cuenta. Sin embargo, se está dejando de lado el riesgo que puede estar asociado a esa ruta. El camino de menor tiempo esperado, puede tener un riesgo que lo hace menos atractivo que otro con tiempo promedio mayor. O incluso, este camino puede tener un peor escenario (por ejemplo el 10 % de los casos) que simplemente provoque que deba descartarse este camino, según los criterios que tenga CBS para acudir a una emergencia.

Algunos de estos estándares pueden definir distintos caminos para llegar a una emergencia. Se puede fijar como límite que en menos de 3 minutos los vehículos estén en el siniestro, con una probabilidad del 95 % (o equivalentemente que el 95 % de los casos estén en este tiempo). O encontrar el camino que tome menos tiempo sugeto a un riesgo o varianza determinados. En cualquiera de estos casos es necesario modelar el tiempo de viaje como una variable aleatoria con distribución conocida para distintos segmentos de un grafo, y posteriormente, descubrir la manera de relacionar estas variables para estimar la distribución de una camino.

El estudio que aquí se desarrolla, busca proponer un método para estimar distribuciones de tiempos de viaje para caminos arbitrarios por un grafo dirigido, lo que permitirá, en otros estudios, plantear modelos de ruteo robustos para caminos mínimos.

1.1. Objetivo general

Definir un método para procesar datos operacionales del sistema de transporte Transantiago, y usar estos datos para representar la incertidumbre de los tiempos de viaje.

1.2. Objetivos específicos

- Desarrollar un modelo de datos y establecer un sistema de almacenamiento y gestión de ellos.
- Proponer un método para identificar recorridos que siguen los buses que producen los puntos GPS.
- Determinar algoritmos de proyección de los datos, y calcular velocidades y tiempos de viaje.

- Identificar la distribución que siguen los tiempos de viaje, y estimar los parámetros que sean necesarios a nivel desagregado.
- Proponer un método para el comportamiento conjunto de los arcos. Encontrar distribuciones para un camino dentro del grafo.
- Definir grupos horarios y similitudes en las distribuciones.

1.3. Hipótesis

Se plantea que existen modelos que permite explicar tiempos de viaje, los cuales puede tratarse a nivel desagregado. En otras palabras, se pueden modelar estas variables aleatorias en segmentos arbitrarios representados como arcos de un grafo. La desagregación puede ser también en el tiempo, ya sean horas o minutos.

Los datos necesarios para estudiar estos modelos se pueden obtener de la base de datos operacional del sistema de transporte urbano de Santiago. Los registros se asume que se encuentran validados, por lo que no es un requisito procesarlos para evitar errores en los resultados.

El comportamiento a nivel agregado de los tiempos de viaje, por ejemplo, a través de un camino cualquiera en el grafo, puede ser evaluado usando parámetros que ya han sido estimados a nivel desagregado.

1.4. Alcances

Desarrollar un método para el procesamiento de datos, desde la data como puntos GPS, hasta distribuciones de tiempos de viaje arbitrarios por Santiago, es un problema complejo. Producto de ello, se dará mayor atención al modelamiento de la incertidumbre y se propondrá, sin mayor profundidad, un método para obtener datos de tiempos. Existen bastantes estudios que abordan el problema de identificar que camino siguió un set de puntos GPS, los cuales en este trabajo no se analizan detalladamente.

Además, se cuenta para cada día con más de 12 millones de registros, asociados a más de 300 recorridos distintos. Un recorrido, a su vez, tiene varios buses que lo transitan, produciendo un punto GPS dentro de la base de datos cada 30 segundos. Existen al menos 6700 buses distintos que producen estos datos. En total, se tienen más de 360 millones de registros correspondientes a un mes completo de funcionamiento del Transantiago. Resulta muy difícil poder procesar y analizar toda esta información, por lo mismo, se desarrollará el método utilizando un subconjunto de estos datos y se instalará la base de datos en un repositorio para que se pueda continuar con la recolección de datos.

Finalmente, se pretende presentar un modelo para la incertidumbre que permita explicar razonablemente bien los tiempos de viaje. Extensiones de estos modelos que mejorarían los ajustes, no se desarrollarán y quedarán propuestos como trabajos futuros.

1.5. Estructura del trabajo

Este trabajo comienza analizando y estudiando el cómo gestionar y almacenar los datos, lo cual se aborda en el Capítulo 2. Posteriormente, se presenta una metodología para proyectar puntos GPS y así identificar el camino que sigue cada recorrido, lo que se lleva a cabo en el Capítulo 3. En el Capítulo 4 se estudia el cómo generar tiempos de viaje y velocidades a partir de los datos, los que luego se discuten en el Capítulo 5. Los resultados más interesantes están en el Capítulo 6, dejando las conclusiones y trabajos futuros para el final.

Capítulo 2

Manejo de datos

Se cuenta inicialmente con dos importantes fuentes de información:

- El mapa de Santiago

- Todos los datos de monitoreo de buses del transporte público de Santiago de un mes completo, correspondiente a junio del 2010.

El mapa está representado como un conjunto de nodos, o simplemente puntos con coordenadas, y arcos que unen estos puntos. El volumen de datos para este grafo no es difícil de manejar puesto que son solo 350 mil nodos y 650 mil arcos. Esta información está originalmente en formato csv, y alcanza los 150MB. Sin embargo, la segunda fuente de información, los movimientos de los buses, es realmente complicada de gestionar. Esta data almacenada en txt supera los 20GB en tamaño en disco, lo cual, no solo es difícil de gestionar, sino que también de mantener. Obtener el camino que siguió un bus determinado implica revisar completamente el archivo del día que se está buscando, e identificar los registros correctos, lo cual es muy ineficiente si lo que se quiere conseguir son datos de tiempos. Por otra parte, en este formato no se tiene certeza de si los datos están completos o no, o de si están duplicados.

Cada registro de estos enormes archivos, originados por tecnología de GPS, tiene los siguientes campos:

1. `patente`: Campo que indica la patente del bus que generó el dato
2. `fecha_hora_gps`: Fecha, hora, minutos y segundos a los cuales se tomó el dato. El formato es YYYYMMDDHH24MISS (año-mes-día-hora(24)-minutos-segundos).
3. `latitud`: Latitud del punto GPS
4. `longitud`: Longitud del punto GPS
5. `ruta_mtc`: Código identificador del recorrido
6. `sentido`: Sentido en el que se encuentra viajando el bus, ya sea ida (1) o vuelta (2).

7. `velocidad`: Velocidad instantánea registrada en ese momento.

Otros sistemas comunes para el manejo de datos como el que se ha descrito, por ejemplo archivos txt, csv, dat, no son capaces de administrar este volumen de información. Es necesario, entonces, buscar alternativas. Una de estas alternativas es un sistema administrador de bases de datos (SABD), lo que almacena sistemáticamente registros para su posterior uso. Existen varias ventajas de mantener un SABD o DBMS (por sus siglas en inglés). *Elmasri* [11] explica cada uno de estos atributos.

1. Control de redundancia: Almacenar los mismos datos varias veces puede conducir a duplicación del esfuerzo, derrochar espacio de almacenamiento e incoherencia de los datos.

2. Almacenamiento persistente: Permite almacenar datos y objetos, que se pueden recuperar cuando se necesiten de la misma forma que se guardaron.

3. Suministro de estructuras de almacenamiento para un procesamiento eficaz de las consultas: Se proporcionan capacidades para ejecutar eficazmente consultas y actualizaciones. Con este fin se crean archivos llamados índices, y se mantiene un módulo de procesamiento y optimización de consultas.

4. Copia de seguridad y recuperación: Se cuenta con un subsistema de copia de seguridad y recuperación, el cual opera en caso de fallas.

5. Representación de relaciones complejas entre los datos: Una base de datos puede incluir numerosas variedades de datos que se interrelacionan entre sí de muchas formas.

6. Implementación de las restricciones de integridad: Permite mantener en el tiempo un sistema con registros que no se repiten, y que deber estar relacionados con otros .

7. Inferencia y acciones usando reglas: Algunos sistemas ofrecen la posibilidad de definir reglas de deducción para inferir información nueva a partir de los hechos guardados en la base de datos. Además permiten implementar procedimientos almacenados.

Un proyecto, como el que se está desarrollando, aprovecha varias de estas ventajas. Una de las más destacables es la que se menciona en el punto 6. la integridad de los datos. Al incorporar restricciones de unicidad de los datos, se garantiza que éstos no se repetirán. Gran parte de los datos que se almacenan tienen valor estadístico, por lo que en caso de que se repitan, podría llevar a conclusiones o estimaciones incorrectas. El punto 2. también ayuda a mantener los datos consistentes, pues al usar un sistema DBMS se tiene seguridad en que los datos no cambiarán a menos que alguien intervenga directamente.

Aproximadamente 12 millones de registros diarios para un mes completo se deben almacenar y recuperar rápidamente. Para filtrarlos se usan distintos criterios de búsqueda, para lo cual se usan los índices. Éstos se mencionan en el punto 3.

Los índices son estructuras de datos, que pueden ser árboles binarios o tablas hash. Su función es procurar que los datos se ordenen y se mantengan de manera que optimizan los tiempos de obtención de éstos. Este factor es clave al momento de tratar con estos casi 360

millones de datos que definen el trayecto de todos los buses para el mes que se tiene.

2.1. Modelo entidad relación

Primero se crea un mapa conceptual de cómo se relacionarán los distintos datos, los cuales se presentan como entidades o relaciones entre éstas. En la Figura 2.1 se expone el diagrama de la base de datos. Existen 4 entidades principales: nodo, arco, recorrido y dato, la última es débil, o en otras palabras, necesita que un recorrido la defina. La entidad dato es simplemente una fila de los archivos de geolocalización de buses del Transantiago, siendo cada una única y definida por una patente, localización, una fecha y hora (tiempo), un recorrido y su sentido.

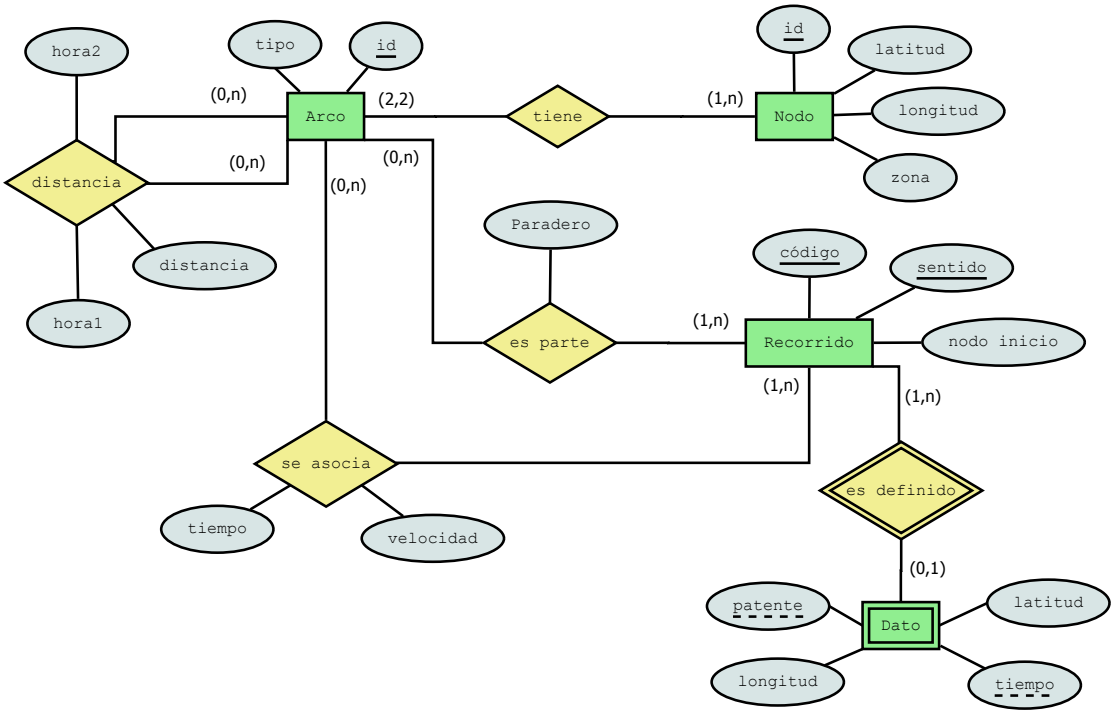


Figura 2.1: Modelo entidad relación que conceptualiza la base de datos

Por otra parte, el grafo de Santiago se representa con nodos y arcos. Cada uno de ellos tiene un `id` que lo define únicamente, ya sean estos arcos o nodos. Los primeros, además cuentan con un entero que define su tipo de calle, es decir, si es una calle, una avenida, una carretera, etc. Dentro de los atributos de los nodos, por otra parte, están su localización (latitud y longitud) y la zona del mapa a la que pertenecen.

Cada nodo puede ser parte de muchos arcos, mientras que cada arco obligatoriamente debe tener dos nodos.

Paralelamente, se conceptualizan los recorridos de los buses. Una línea tiene un único par código y sentido, lo que en conjunto permite conocer cuál es el nodo dónde comienza un

recorrido para un sentido dado. De esta manera, es posible conocer cada punto extremo de un recorrido, pues es nodo inicio de la ida o la vuelta del bus. Cuando se combina esta entidad con los arcos, se tiene el conjunto de arcos que define un recorrido. El arco al cual se le hace referencia puede ser un paradero o no, lo que se indica con un 0 o un 1.

Finalmente, hay dos relaciones que tienen especial importancia para el análisis estadístico. Se trata de la relación arco-recorrido que tiene una velocidad y la asociación entre dos arcos. La primera tiene la velocidad y momento con que un bus atravesó un arco determinado, lo que permite entender la distribución que tiene esta variable aleatoria y el tiempo de viaje en el mapa, mientras que la segunda, otorga las diferencias entre las distribuciones de un par (arco,hora) distintos, haciendo posible la segmentación agrupación de distribuciones.

2.2. Modelo relacional

Cuando el modelo entidad relación está definido, se da paso a la construcción del modelo relacional, o en palabras simples, las tablas de la base. Se recomienda seguir una serie de pasos para obtener este modelo, los cuales son descritos por *Elmasri* [11]. Siguiéndolos, y ocupando una serie de reglas de normalización que también se especifican, se propone el siguiente set de tuplas:

- `nodo(id,latitud,longitud,zona)`
- `arco(id,nodo_inicio,nodo_fin,tipo)`
  `Llave foránea (nodo_inicio) ref nodo, (nodo_fin) ref nodo`
- `recorrido(código,sentido,nodo)`
- `arco_rec(id,código,sentido,paradero)`
  `Llave foránea (id) ref arco, (código,sentido) ref recorrido`
- `trayecto(id,código,sentido,tiempo,velocidad)`
  `Llave foránea (id) ref arco, (código,sentido) ref recorrido`
- `distancia_arco(id1,hora1,id2,hora2,distancia)`
  `Llave foránea (id1) ref arco, (id2) ref arco`
- `data(patente,fechahora,codigo,sentido,latitud,longitud)`
  `Llave foránea (codigo,sentido) ref recorrido`

El subrayado de un campo o varios de ellos, indica que éste o el conjunto de estos, son la llave primaria de la tabla (conjunto de valores mínimos que son únicos a cada registro).

En definitiva, se consiguen 7 tablas, 3 de las cuales provienen directamente de los datos (data, arco y nodo), y las otras 4 requieren de algoritmos para construirse.

2.3. Índices

Existen varias alternativas para indexar una base de datos. Conocidos son los árboles de búsqueda binaria, siendo una generalización de estos el B^+ tree, el cual hace posible tanto buscar como insertar y borrar datos con eficiencia garantizada en el peor caso. Estos árboles son descritos por *Knuth* [15] como una estructura de datos que permite realizar estas operaciones en tiempo logarítmico, y que a diferencia de los árboles binarios, pueden tener más de dos hijos desde un mismo nodo. Estas estructuras permiten encontrar tanto un valor preciso que tenga una fila de la tabla como un rango que se especifique.

Por otro lado, un índice que también es muy usado es el hash, el cual usa un argumento K. Este factor es una clave de la tabla que se usa para calcular una función $f(K)$ que dará la localización en memoria del registro con campo identificador K en la tabla [15]. A pesar de que no es fácil encontrar tal función, este método permite búsquedas o modificaciones muy eficiente para condiciones con igualdad. Es decir, siempre que se quiera buscar un campo con una llave única, como un id, este índice tomará un tiempo constante, $\mathcal{O}(1)$.

Lo primero que se debe hacer es indexar la tabla de arcos con un hash, puesto que cada fila está únicamente definida por el campo `id`. Cada vez que se quiera buscar un arco, este método lo hará de forma eficiente. Además, debido a que cada `id` se coloca de forma arbitraria para un arco, no se realizarán búsquedas por rango, por lo que no se justifica un B^+ tree. Asimismo, los nodos tendrán un índice hash para acceder a ellos. Sin embargo, se decide incluir más de un índice en la tabla nodo. La latitud y la longitud de cada punto, sí tiene sentido que se incluya un método para búsqueda por rango, por lo tanto, aquí se usa un B^+ tree para estos campos.

Finalmente, se usa un índice agrupado para la tabla data. De estos datos saldrán las velocidades, haciendo que la búsqueda en esta tabla deba ser muy eficiente. Puesto que el cálculo de velocidades se hace por recorrido, el campo que se indexa es ese, el que está representado por `código`. A este campo, por consiguiente, se le coloca un B^+ tree.

Se prueban estos índices en la tabla `data` con aproximadamente 60 millones de registros. Como se mencionó anteriormente, esta tabla ya tiene un índice en la columna `código`, atributo que se usará para filtrar. Ante la consulta:

```
SELECT *
FROM data
WHERE codigo = 'C11'
```

el resultado sin índices tarda un tiempo promedio de 126.16 segundos, mientras que con índices este promedio es mucho más bajo, llegando a los 3.34 segundos.

2.4. Instalación en servidores

La base de datos completa está albergada en un clúster o conjunto de computadores instalado en dependencias del Departamento de Ingeniería Industrial de la Universidad de Chile. El sistema está administrado por una distribución de linux, y gracias a su diseño permite procesar trabajos que requieren muchos recursos sin problema. Actualmente, ya se han cargado varios datos que se encontraban en formato txt los que tienen relación con las posiciones de los vehículos. Esto ha permitido organizar de mejor manera los casi 360 millones de registros que componen el total. Además, las tablas que son pobladas mediante el procesamiento de estos datos, han comenzado a ocuparse. La cantidad de datos para cada tabla se informa en la Tabla 2.1.

	nodo	arco	recorrido	arco_rec	trayecto	distancia_arco	data
Cantidad	325.262	662.743	563	7.659	3.013.312	29.610.360	122.987.378

Tabla 2.1: Cantidad de datos para cada tabla de la base de datos

Capítulo 3

Definir rutas para cada recorrido

3.1. Reducción del grafo

Los datos GPS contienen todos los movimientos de los buses, ya sea en su recorrido definido, cuando se desvían producto a algún inconveniente e incluso cuando no están operando. Muchos de estos datos, por lo tanto, no se sabe que camino realmente tomaron o bajo qué condiciones fueron emitidos, lo cual podría eventualmente ensuciar la base de datos de velocidades y tiempos de viaje. Es por ello que en una primera fase, antes de proyectar estos datos, se identifica cuál es el camino en el grafo del servicio, por donde se sabe que el bus debe pasar. Si los puntos se alejan considerablemente de este camino, entonces no se deben tomar en cuenta. Gracias a esto, en caso de que los puntos tengan algún error, se tendrá garantía de que siempre serán los mismos arcos dónde se proyectarán. Además, es fácil implementar criterios para definir si un bus abandonó la ruta o no.

Sin embargo, el grafo de Santiago tiene más de 350 mil nodos y 650 mil arcos. Utilizar el grafo completo para un recorrido que solo recorre un pequeño set de arcos, simplemente, no tiene sentido. Lo ideal es solo usar aquellos arcos que están cerca de los puntos GPS que se quieren mapear. Así, entonces, inicialmente se selecciona la mínima cantidad de arcos candidatos cercanos a los puntos, sin perder los arcos donde realmente transitan los vehículos.

Se comienza eligiendo los últimos 150 viajes desde el punto de inicio hasta el punto de fin de un recorrido, en un sentido determinado, ya sea ida (1) o vuelta (2). Estos viajes son obtenidos directamente de la base de datos que se analizó en el Capítulo 2. La tabla `data` tiene por cada fila una transacción de punto GPS. El método pretende escoger la versión más actual que sigue este recorrido, por lo mismo, es que se toman los últimos viajes que siguieron buses de esta línea. Es común que los caminos seguidos por estos buses tengan modificaciones a lo largo del tiempo, lo que se busca incorporar para estimar correctamente tiempos y velocidades de viaje. Si el camino que se tiene para proyectar no corresponde a los puntos GPS, entonces muchos datos serán inconsistentes. Idealmente, periódicamente los caminos recorridos deberían actualizarse.

Como criterio se seleccionan solo puntos que pertenecen a una misma patente, y que se

están moviendo en un intervalo de tiempo. Buses que emiten durante un tiempo prolongado puntos GPS con la mismas coordenadas quedan fueran del análisis, ya que probablemente están fuera de servicio. Cabe destacar que dentro de los datos suministrados, se hayan los registros que los buses emiten durante la noche y madrugada, cuando están inactivos. Estos datos se deben excluir, pues no sirven ni para definir los recorridos, ni para estimar velocidades.

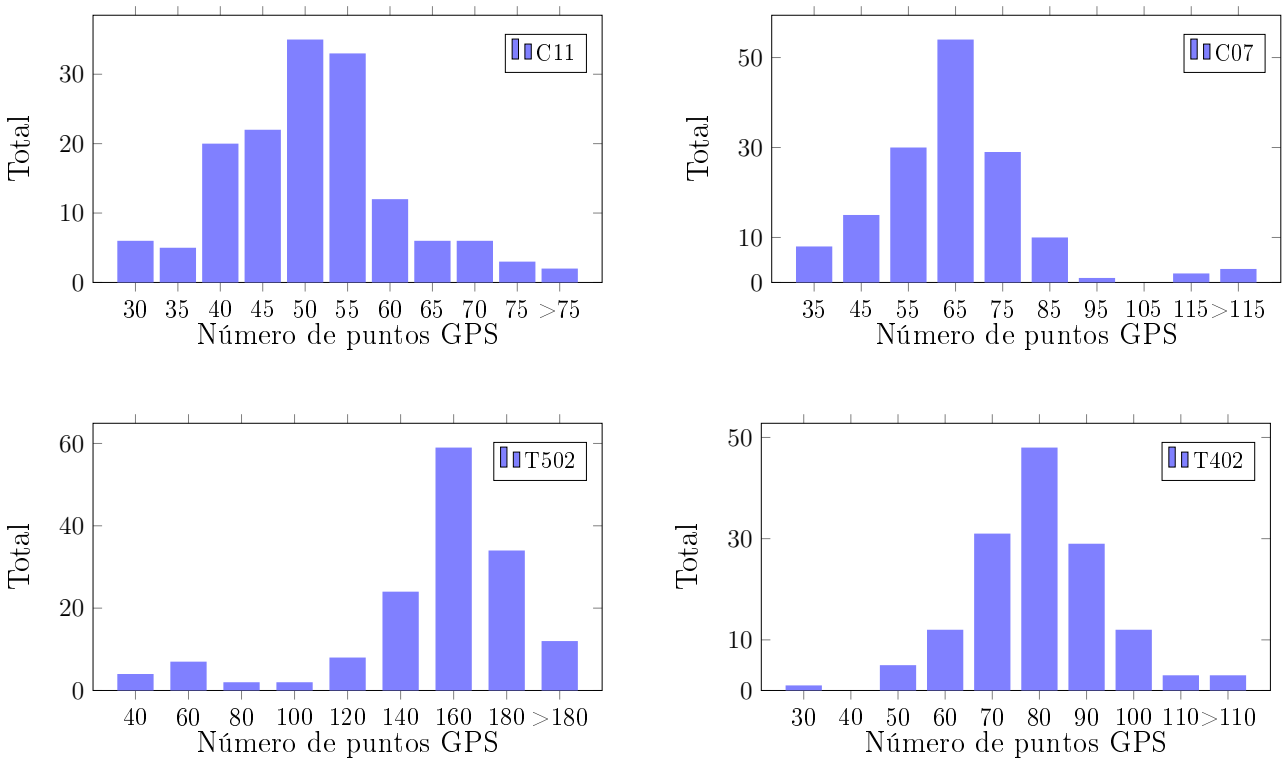


Figura 3.1: Histogramas del número de puntos GPS de cada viaje para ejemplos de recorridos

Los 150 viajes tienen una distribución como la que se despliega en la Figura 3.1. Aquí se grafican como ejemplo cuatro líneas distintas, con recorridos y patrones distintos. Queda claro que para definir que camino sigue habitualmente la linea no se deberían usar las muestras ni de un extremo de la distribución, ni del otro, probablemente estos sean rutas distintas o circuitos incompletos. Si por ejemplo, se identificara un camino con solo 10 puntos GPS, para un recorrido que cruza la ciudad, aunque viajara a una gran velocidad, no podría emitir tan pocos puntos para completar el camino.

Por esta razón, se deben usar las muestras que se encuentran en la mitad de la distribución. Por consiguiente, para seleccionar el subgrafo, los puntos GPS que lo generan son aquellos que pertenecen al set de puntos entre 50 y 100 del histograma. Lo que se pretende es usar solo muestras que estén correctas. Con estos datos se busca identificar el camino que deberían seguir los buses, por lo que éstos deben ser lo más representativos posible. Los datos de la parte media de estos histogramas debería responder con ese requisito.

El subgrafo que se quiere armar pretende simplificar el trabajo de identificar los arcos para cada línea. Es bastante más fácil encontrar el camino que ha seguido un bus en solo 10 mil arcos que en los 650 mil que tiene el mapa de Santiago. Con esto además se facilita la

proyección que se debe hacer de cada punto y también se reducen las opciones de caminos que siguió cada bus en el mapa.

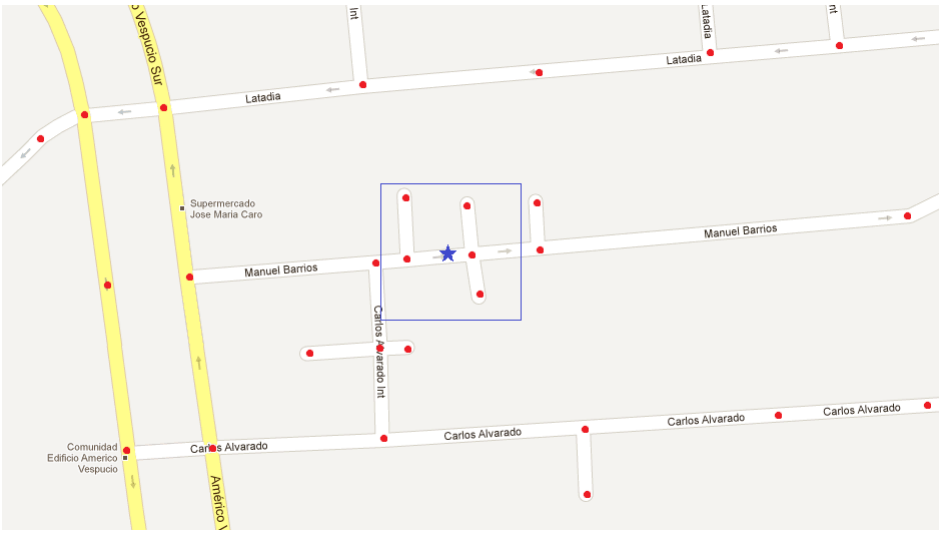


Figura 3.2: Ejemplo de selección de nodos entorno a un punto GPS

Para cada uno de estos puntos pertenecientes a los viajes entre la posición 50 y 100 del histograma, se miran los puntos del grafo que están en un radio entorno a éste. En la Figura 3.2 se ejemplifica este procedimiento. Se toma un punto, y se seleccionan como candidatos todos aquellos nodos a una distancia dada. La distancia que se calibró, con la cual era posible generar el subgrafo necesario, es de 40 metros. Así se garantiza que en la medida que los buses pasen al menos a 40 metros de los arcos que le corresponden, el camino que siguen será identificado. Se consigue, entonces, el objetivo planteado: seleccionar solo una parte del grafo, la mínima posible para definir una ruta.

Un recorrido en particular, genera el subgrafo que se observa en la Figura 3.3. Del mapa completo de Santiago, se selecciona este subgrupo el cual contiene el camino para este recorrido, con bastante seguridad. Luego, basta con mirar este reducido grupo para aplicar los algoritmos de map matching, lo que simplifica los tiempos de proceso y disminuye los errores.

3.1.1. Estructura de datos para un subgrafo

Una vez que se tiene el subgrafo, éste se guarda en el sistema formado por una estructura de datos específica, la cual representa un objecto subgrafo. La unidad básica de esta estructura es un arco, es decir, cada subgrafo está constituido en su origen por un arco. Cada arco tiene los siguientes atributos:

- id que lo caracteriza, el mismo de la tabla `arco` de la base de datos.
- id del nodo de inicio del segmento y del nodo de fin.
- coordenadas de estos nodos, lo que posibilita ubicar el arco.
- una lista con todos los arcos que salen de este arco. En otras palabras, un conjunto que tiene los arcos que tienen como nodo de inicio el nodo fin de este arco.

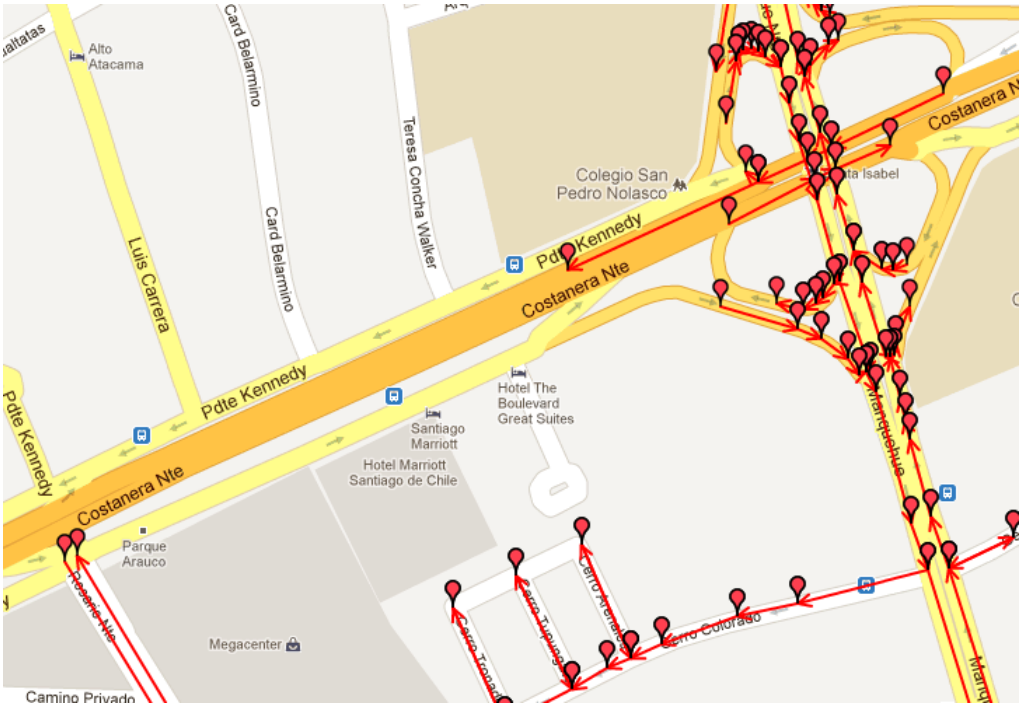


Figura 3.3: Segmento del subgrafo que genera el recorrido C11

Este sistema ayuda a que se conozca de manera muy directa un camino de arcos.

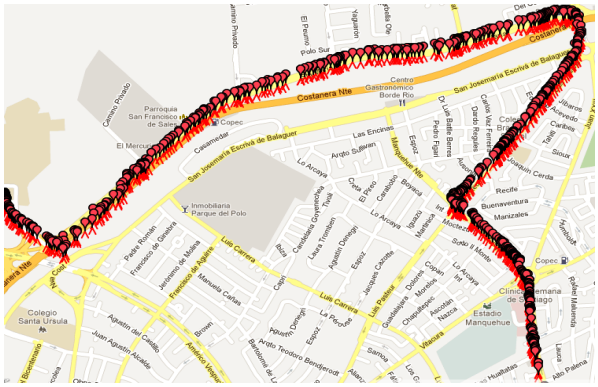
Cada subgrafo tiene estos arcos almacenados en una tabla hash, es decir, en una tabla que guarda una llave para cada entrada, lo que da acceso rápido al registro. Esta estructura mapea un valor único para cada registro (en este caso el id del arco) con una dirección de memoria dónde se ubican los atributos asociados a este id. Para ello utiliza una función hash [15].

Sin embargo, eso no es suficiente. Se requiere que rápidamente se pueda acceder a todos los arcos que estén cerca de un punto GPS, con el objetivo de que sea fácil proyectarlo en este grafo. Por lo mismo, se cuenta con una lista que indexa el grafo por su latitud, lo cual permite dividirlo en pedazos que son de igual medida. El grafo finalmente, se guarda en el sistema como una lista enlazada, con "celdas" de arcos.

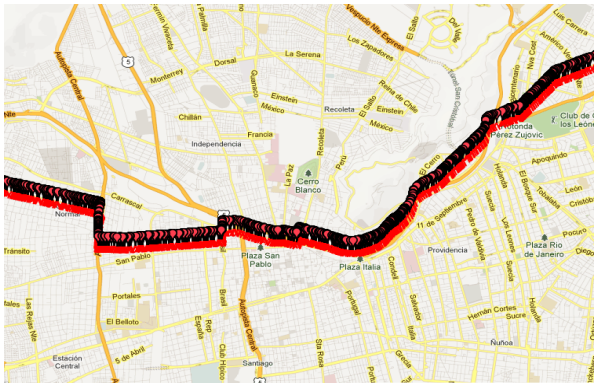
Se estimó que una distancia de 150 metros para estas celdas es adecuada para proyectar correctamente todos los puntos.

En la Figura 3.4 se muestran los puntos que posteriormente ayudarán a identificar las rutas para un recorrido en particular. Lo que se está graficando aquí es el set de puntos GPS de los caminos más representativos del recorrido, es decir, se han tomado distintas muestras de caminos que han seguido buses de este recorrido, y se grafican en el mapa. Se aprecia que el proceso de selección de puntos con gran seguridad permite representar los arcos del grafo que se han usado.

En este punto, nuevamente se debe considerar el gráfico de la Figura 3.1. En efecto, del total de caminos, 150 para cada recorrido y sentido, se eligen aquellos que están en la moda



(a) Puntos GPS del recorrido C07 en sentido ida



(b) Puntos GPS del recorrido T502 en sentido vuelta

Figura 3.4: Ejemplo de puntos GPS que se seleccionaron para un recorrido

del total de puntos, en otras palabras, aquellos que al ordenarlos, están en la posición 70 hasta la 80, ambas incluidas. Así como antes se tomaron los 50 más representativos para dar origen al subgrafo (el cual evita tener que revisar el mapa completo de Santiago), para proyectar, se usan los 12 que probablemente han seguido el camino correctamente.

Lo que se está haciendo es escoger caminos de puntos GPS que permitan, de la forma más fidedigna, obtener el camino buscado. Luego, se toman estos 12 set de puntos y se les aplica el método que se explica a continuación.

3.2. Modelo de map matching

El primer objetivo que se plantea es definir los recorridos que siguen los puntos GPS. Alrededor de 300 patrones de rutas están implícitos en la data entregada por Transantiago, los cuales deben ser mapeados al grafo que se tiene de Santiago. Cada uno de estos recorridos, además, tiene dos sentidos lo que conduce a un gran número de caminos que identificar. Con los caminos encontrados, posteriormente, se estiman las velocidades y, consecuentemente, los tiempos de viaje que definen los datos de posicionamiento.

Los datos entregados contienen, además de los puntos de posicionamiento de los vehículos, las rutas de los recorridos usando polilíneas de 500 metros. En una fase anterior del proyecto se hizo uso de esta fuente de información para estimar tiempos para el algoritmo de caminos mínimos implementado. Además, el uso de estos datos podría simplificar la identificación de recorridos con el mapa, pues habría que proyectar una menor cantidad de puntos, los que además tendrían garantías de ser correctos. Sin embargo, se decide no utilizarlos ya que el sistema que se describe en este capítulo permite independizarse de esta entrada y, además, actualizar los caminos automáticamente en caso de que estos cambien en el tiempo. Cada vez que se ingresen nuevos datos de posicionamiento, el sistema podrá actualizar los recorridos de forma automática, y así estimar datos de viaje con información confiable.

El problema de map matching o identificación de caminos ha sido ampliamente estudiado,

y simplemente corresponde a determinar que camino está usando un vehículo que emite coordenadas de ubicación en un grafo dirigido.

El algoritmo más simple sería proyectar cada uno de los puntos que se han seleccionado en el arco más cercano posible, y luego conectar estos arcos. El principal problema de este método es que puede fallar enormemente si los puntos tienen errores, como es el caso de los datos de buses que se está usando. *White et al.* [32] estudió este simple algoritmo y dos versiones con ajustes. Agregó información en la orientación de los arcos y para la siguiente versión, además, incluyó restricciones de conectividad entre los arcos escogidos. Estos algoritmos puramente geométricos los comparó con uno que realizaba ajustes entre las curvas, encontrando que esta última versión, más sofisticada que las anteriores, tenía un peor desempeño que uno de los enfoques geométricos.

Mientras algunos autores han estudiado caminos factibles usando errores ponderados, otros se han enfocado en la forma de estos caminos, aunque todos ellos buscan disminuir el error existente entre el camino sugerido y el real. Sin embargo, algoritmos que simultáneamente disminuyan el error y consideren varias hipótesis de caminos, no son la mayoría. *Newson et al.* [20], proponen un algoritmo que toma en cuenta estos dos puntos, justificando que un problema potencial de algoritmos solamente geométricos es que son muy sensibles a ruido en las mediciones y distancia entre cada uno de los puntos. Ellos sugieren un modelo que identifica el camino más probable dados los puntos GPS que se quieren analizar. El conjunto de arcos que se identifican a partir de los GPS, es aquel que según criterios geométricos y factibilidad del camino, es el que fue seguido por el vehículo con mayor probabilidad.

El principal problema en map matching, es que la decisión se toma en un tradeoff: escoger arcos cercanos a los puntos o sugerir caminos factibles. En un extremo, si solo se usara como indicador el error con el arco para cada GPS, las rutas sugeridas involucrarían, probablemente, extraños loops, giros en U, etc. Por ejemplo, en la Figura 3.5 se muestra un camino que sigue un vehículo, y los posibles arcos dónde se pueden proyectar las coordenadas que genera. Si en este caso, los puntos se proyectaran en los caminos más cercanos, claramente la ruta que se formaría no tendría sentido y debería realizar extrañas vueltas para poder conectarse. Por otra parte, si se toma en consideración solo caminos factibles la cantidad de combinaciones posible vuelve mucho más lento el algoritmo. Para evitar caminos extraños y manejar una cantidad razonable de alternativas, *Newson et al.* [20] incorporan criterios para que el camino encontrado sea verosímil y tenga parecido a la secuencia que se está observando.

Su trabajo se basa en Hidden Markov Model (HMM), es decir en cadenas a las cuales se les conoce parte de la información, mientras que otra parte se encuentra "oculta", o que simplemente donde no se conoce el estado en que se encuentra el sistema. Este enfoque resuelve el problema de conectividad entre los arcos y considera, al mismo tiempo, varias hipótesis de caminos que se pueden armar con estos arcos. Además, incluye criterios geométricos para las proyecciones de los puntos GPS.

El modelo de HMM considera un camino en un grafo como transiciones entre varios posibles estados, donde algunas de estas transiciones son más probables que otras y donde además, el estado que se está analizando es incierto.

Para este modelo, los estados de la HMM son los segmentos individuales del camino, es

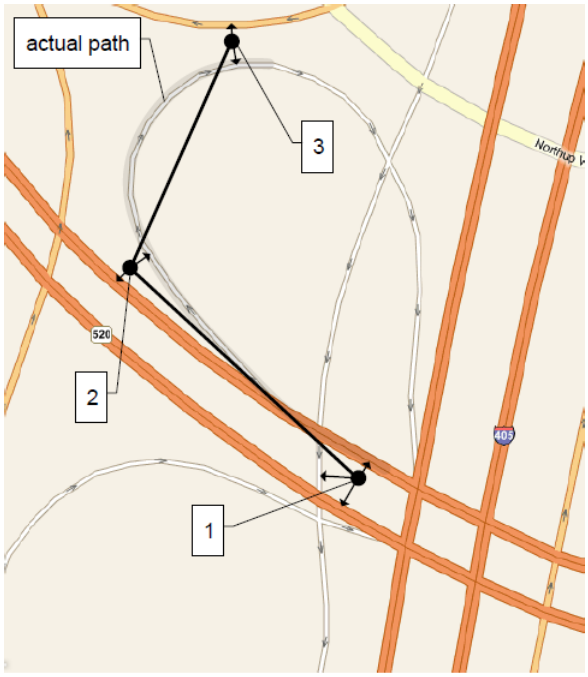


Figura 3.5: Ejemplo de una ruta y los arcos dónde se podrían proyectar puntos GPS

decir, los arcos del mapa. Estos estados de la cadena corresponden a la parte que se puede observar y donde es posible conocer la transición de un estado a otro (qué tan probable es pasar de un arco a otro). En este caso las transiciones entre estados dependen de la conectividad de los arcos. Los estados ocultos o estados medidos, son los puntos GPS con sus respectivas latitud y longitud. Un punto GPS puede estar asociado a cualquier arco, por lo que su "real" estado no se conoce. La meta es asociar cada posición GPS con el camino apropiado según cercanía de los puntos al camino y la factibilidad de éste, lo cual está simultáneamente incorporado en el modelo.

Sean g_{i}, $\mathrm{i} \in S$ puntos GPS determinados por una latitud y longitud, estos son los estados medidos. Además, sean a_j, $j \in N$ arcos del grafo los cuales forman parte de los estados discretos de la HMM. Finalmente, la proyección de un punto GPS i sobre el arco j se le llamará $p_{\mathrm{i}j}$, el cual se define por una latitud y longitud dentro del segmento que define el arco j. Esta proyección se lleva a cabo calculando los siguientes valores:

Sea un arco de nodo inicio (x_1, y_1) y un nodo fin (x_2, y_2), y un punto GPS de coordenadas (a, b). Entonces, la proyección de este punto sobre el arco, proyección que se llamará $(\bar{a}, \bar{b})$, viene dada por:

$$\bar{a} = (x_2 - x_1)\lambda + x_1 \tag{3.1}$$

$$\bar{b} = (y_2 - y_1)\lambda + y_1 \tag{3.2}$$

$$\lambda = \frac{(x_2 - x_1)(a - x_1) + (y_2 - y_1)(b - y_1)}{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3.3}$$

Se puede modelar la distancia entre $p_{\mathrm{i}j}$ y g_{i}, dado un arco j, como una variable aleatoria

normal de media cero y varianza σ^2 [20]. Esta distribución es escogida por los autores, pues establecen que los errores que tienen los datos son normales centrados en 0, entorno al camino que siguen los vehículos que emiten GPS. Así, la probabilidad que un punto g_i pertenezca al arco a_j es

$$\Pr(g_\mathrm{i}|a_j) = \frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{\left(-0{,}5\frac{||g_\mathrm{i}-p_{\mathrm{i}j}||_{\text{great circle}}}{\sigma}\right)^2} \tag{3.4}$$

La distancia entre puntos GPS, $||\cdot||_{\text{great circle}}$, no es la distancia conocida de norma dos, sino una con una expresión un poco más complicada. Sea x_1, y_1 latitud y longitud en radianes del punto 1, y x_2, y_2 del punto 2. La distancia d entre estos puntos corresponde a

$$||(x_1, y_1) - (x_2, y_2)||_{\text{great circle}} = 2R \cdot \arctan 2(\sqrt{\alpha}, \sqrt{1-\alpha}) \tag{3.5}$$

donde el valor de a es

$$\alpha = \sin^2\left(\frac{x_2 - x_1}{2}\right) + \cos(x_1)\cos(x_2)\sin^2\left(\frac{y_2 - y_1}{2}\right) \tag{3.6}$$

A estas probabilidades, $Pr(g_\mathrm{i}|a_j)\ \forall \mathrm{i}, j$, se les conoce como measurement probabilities y son la "manifestación" de las observaciones que ayudan a descubrir los estados de la cadena. Permiten estimar la probabilidad de que un punto GPS pertenezca a un segmento, basado solamente en la latitud y longitud. Reflejan una idea que es clave, para un punto g_i y un arco a_j, la probabilidad de que este punto haya sido producido por un vehículo en a_j, debe disminuir en la medida que la distancia proyectada aumenta.

Para la implementación de este algoritmo se fijó que 30 metros entre la proyección de un punto y un arco, era lo máximo permitido. Los candidatos a arcos de un recorrido disminuyen notablemente, sin perjuicio de la calidad de los resultados, puesto que los puntos GPS con que se trabaja son de alta precisión. Para los casos observados, la mayoría de las proyecciones estaban a menos de 5 o 10 metros del arco que estaba utilizando el bus.

Por otra parte, se debe establecer la probabilidad de que, una vez que se conocen los arcos dónde se proyectan los puntos GPS, la secuencia definida entre estos arcos sea la correcta. A estas probabilidades en la cadena se les llama transition probabilities, y corresponden a las transiciones entre estados discretos de HMM. Estas probabilidades se definen entre estados conocidos, a diferencia de las measurement probabilities.

Sea un punto g_i y su sucesor $g_{\mathrm{i}+1}$ de una secuencia de puntos que haya realizado un vehículo. Estos puntos se proyectan sobre distintos arcos, siendo $p_{\mathrm{i}j}$ el punto sobre el arco j del punto i. De la misma forma, $p_{(\mathrm{i}+1)k}$, es el sucesor proyectado sobre k. La distancia entre estos dos arcos, siguiendo la ruta mínima entre ellos se le denotará $||\cdot||_{\text{ruta}}$, distancia que se compara con la que tienen los puntos g_i y $g_{\mathrm{i}+1}$, $||g_\mathrm{i} - g_{\mathrm{i}+1}||_{\text{great circle}}$. Experimentos revelan que es común que estas dos distancias coincidan, por lo mismo, diferencias menores indican que con mayor probabilidad estos son los arcos para el par de puntos GPS que se está mirando.

Se utilizará el algoritmo de *Dijkstra* para determinar la distancia mínima entre dos arcos. Esta implementación tiene una modificación, se usarán solo los nodos que estén a una distancia específica del nodo de inicio. No es muy probable que el camino real supere mucho en distancia el camino que encuentre *Dijkstra*.

Se justifica esa decisión, ya que la probabilidad de pasar de un arco a otro en el modelo, disminuye bastante cuando la distancia en los arcos es mayor a 3 veces la distancia que definen los puntos de inicio y fin. Con esto, lo que se está haciendo es buscar el camino mínimo entre dos puntos, tales que el camino esté contenido en una circunferencia de radio igual a 3 veces la distancia entre los puntos. Además se debe notar que los puntos son generados cada 30 segundos, lo cual los hace estar relativamente cerca uno de otro.

Sea N el conjunto de nodos que pertenecen a la circunferencia que se especificó. El algoritmo de *Dijkstra*, el cual se analiza y detalla en [3], el trabajo original de *Edsger Dijkstra*, es el que se encuentra en detalle en el Algorithm 1.

Algorithm 1: Dijkstra

$S = \emptyset$; $\bar{S} = N$
d(i) $= \infty$ para cada nodo i $\in N$
d$(s) = 0$ and $pred(s) = 0$
while $(|S| < n)$ **do**
    sea i $\in \bar{S}$ un nodo tal que d(i) $=$ mín$\{$d$(j) : j \in \bar{S}\}$
    $S = S \cup \{$i$\}$
    $\bar{S} = \bar{S} - \{$i$\}$
    for *cada* $(i, j) \in A$(i) **do**
        if d$(j) >$ d(i) $+ c_{ij}$ **then**
            d$(j) =$ d(i) $+ c_{ij}$ and $pred(j) =$ i

Cuando ya se han calculado las distancias mínimas, se calcula la diferencia que existe entre estas, es decir:

$$\Delta = |||g_i - g_{i+1}||_{\text{great circle}} - ||x_j - x_k||_{\text{ruta}}| \tag{3.7}$$

Para este cálculo se hace utiliza la función de distancia $||\cdot||_{\text{great circle}}$ y la que permite calcular la distancia en la ruta $||\cdot||_{\text{ruta}}$. Esta función suma el largo de todos los arcos que están dentro de la ruta que se ha seleccionado.

Usando esta diferencia se puede obtener la probabilidad de pasar del arco j al k dados los puntos g_i y g_{i+1}.

La distribución de estas diferencias según [20] es exponencial de parámetro β,

$$\Pr(\Delta) = \frac{1}{\beta} e^{-\frac{\Delta}{\beta}} \tag{3.8}$$

Este parámetro β debe calibrarse de manera de que el modelo encuentre correctamente el camino.

En este punto, ya se tienen definidas las probabilidades tanto de transición, como de pertenecer a un arco dado un punto GPS. Falta especificar las probabilidades del estado inicial de la cadena, los que se denotarán por π_{i} i $\in S$. El conjunto S es el conjunto de los estados de la cadena. Estos π_{i} se podrían fijar arbitrariamente en un subconjunto de arcos, sin embargo, se recomienda utilizar las proyecciones de la primera observación, es decir, $\pi_{\mathrm{i}} = \Pr(g_1|a_{\mathrm{i}})\ \forall \mathrm{i} \in S$.

El modelo HMM necesita dos parámetros que se relacionan con las probabilidades de esta cadena. Uno es σ, el cual es la desviación estándar del ruido Gaussiano de los puntos GPS, el error que mostraban los datos que emiten los vehículos con respecto al camino que realmente siguen. *Neson et al.* estiman este valor usando datos de los cuales conocían el camino que los había producido. Para ello, tomaron la desviación absoluta media de los errores que se presentaban entre los puntos que habían producido y el camino real, lo que les permitió calcular el valor de σ como sigue:

$$\sigma = 1{,}4826 \cdot median_{\mathrm{i}} \left(||g_{\mathrm{i}} - p_{\mathrm{i}j^*}||_{\text{great circle}} \right) \tag{3.9}$$

Otro parámetro necesario es β, el cual es el componente de la probabilidad de transición de un estado a otro. En [20], se sugiere estimar este valor como:

$$\beta = \frac{1}{\ln(2)} \cdot median_{\mathrm{i}} \left(||g_{\mathrm{i}} - g_{\mathrm{i+1}}||_{\text{great circle}} - ||p_{\mathrm{i}j^*} - p_{(\mathrm{i+1})j^*}||_{\text{ruta}} \right) \tag{3.10}$$

Los parámetros σ y β representan el tradeoff entre qué tan confiable son las proyecciones sobre los arcos, y las rutas encontradas. Un número grande para σ significa que se tiene menos confianza en los valores medidos de GPS. Un valor grande para β se traduce en que se da mayor flexibilidad en tomar caminos no directos, es decir, en que el camino se desvíe de la línea que une los puntos GPS.

Se sugieren valores para cada uno de estos parámetros, $\sigma = 5{,}7$ y $\beta = 12{,}5$, lo cuales se calibraron según los datos que se tienen ,los cuales, además, no difieren mayormente de los sugeridos por *Newson et al.* [20].

Tomando en cuenta todos estos factores, la cadena queda completamente definida. Un ejemplo de cómo se vería se encuentra en la Figura 3.6. Aquí los a_{i}, i $= \{0, ..., k\}$ son los estados de la cadena, los que representan a los arcos; p_{i}, i $= \{1, ..., l\}$ son las transition probabilities; ob_{i}, i $= \{1, ..., n\}$ son las observaciones o puntos GPS que se tienen para un vehículo; y q_{i}, i $= \{1, ..., m\}$ son las measurement probabilities.

El objetivo es encontrar la secuencia de arcos o estados de la cadena que son más probables, dadas las observaciones que se tienen. En consecuencia, lo que se busca es maximizar la probabilidad conjunta de las observaciones y las transiciones a lo largo de toda la cadena. El tipo de algoritmo que se utiliza es uno de programación dinámica. Dentro de esta clase, uno

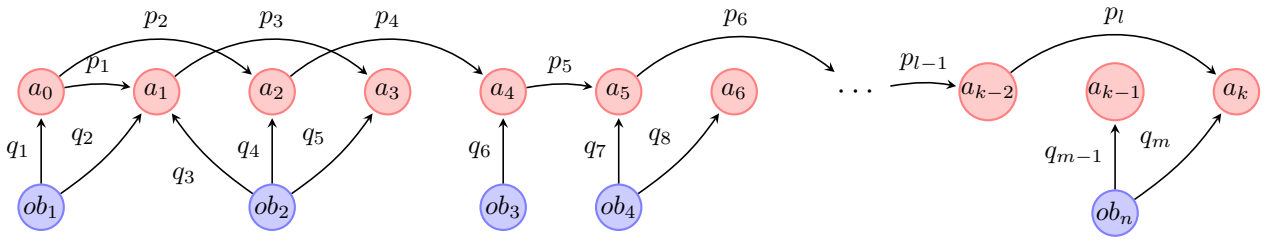


Figura 3.6: Ejemplo de una configuración para Hidden Markov Map Matching

de los más conocidos para resolver HMM, es el algoritmo de *Viterbi*.

El algoritmo de *Viterbi*, es un algoritmo que encuentra los estados más probables para una cadena HM, y además, lo hace en un breve tiempo. La descripción de este algoritmo la publicó *Andrew Viterbi* el año 1967 [29].

Se define el conjunto S como el set de todos los estados, π_{i}, $\mathrm{i} \in S$ probabilidades iniciales, la matriz $A = \{a_{\mathrm{i}j} : (\mathrm{i}, j) \in S^2\}$, matriz de transición, las observaciones y_{i}, $\mathrm{i} = \{1, ..., T\}$ y $B = \{b_{\mathrm{i}y_j} : \mathrm{i} \in S, j = \{1, ..., T\}\}$. El algoritmo se detalla en 2.

Algorithm 2: Viterbi

for *cada estado* s_{i} **do**
  $T_1(\mathrm{i}, 1) = \pi_{\mathrm{i}} b_{\mathrm{i}y_1}$
  $T_2(\mathrm{i}, 1) = 0$
for $\mathrm{i} = 2, 3, ..., T$ **do**
  for *cada estado* s_j **do**
    $T_1(j, \mathrm{i}) = \mathrm{máx}_k\{T_1(k, \mathrm{i}-1) \cdot a_{kj} \cdot b_{jy_{\mathrm{i}}}\}$
    $T_2(j, \mathrm{i}) = argmax_k\{T_1(k, \mathrm{i}-1) \cdot a_{kj} \cdot b_{jy_{\mathrm{i}}}\}$
$z_T = argmax_k\{T_1(k, T)\}$
$x_T = s_{z_T}$
for $\mathrm{i} = T, T-1, ..., 2$ **do**
  $z_{\mathrm{i}-1} = T_2(z_{\mathrm{i}}, \mathrm{i})$
  $x_{\mathrm{i}-1} = s_{z_{\mathrm{i}-1}}$

La complejidad de este algoritmo es $\mathcal{O}(T \times |S|^2)$, y para ser implementado se utilizó la transformación logarítmica de las probabilidades para otorgar estabilidad computacional al algoritmo. Usando las probabilidades sin transformar, se alcanzan a computar alrededor de 40 observaciones, cuando lo común es enfrentar recorridos con más de 100. Las probabilidades que describen la cadena son del orden de 0.1, lo que dificulta bastante el cálculo directo.

Como ya se había mencionado, se tomaron 12 candidatos de posibles caminos. Cada uno de estos entrega el camino de arcos que más probablemente usaron, los arcos por donde pasó el bus que generó estos puntos. Luego, se toman todos estos arcos y se define un criterio para admitirlo dentro del camino de un recorrido. Un arco se debe repetir a lo menos 4 veces de las 12 posibilidades para que pueda ser seleccionado. Con esto, los caminos que toman distintas

rutas, que se desvían o que simplemente alteran el camino, no influyen en el camino principal que se está armando.

En la Figura 3.7 se muestra un ejemplo del camino que finalmente se arma usando estos puntos, el cual, si bien no tiene garantía de que sea 100 % correcto, tiene un respaldo razonable de que es el que se está buscando.

3.3. Recorridos procesados

El procedimiento presentado se siguió para varios recorridos los cuales fueron cargados a la base de datos que se tiene para el proyecto. La tabla que alberga estos resultados es la `arco_rec` que asocia cada arco con cada recorrido.

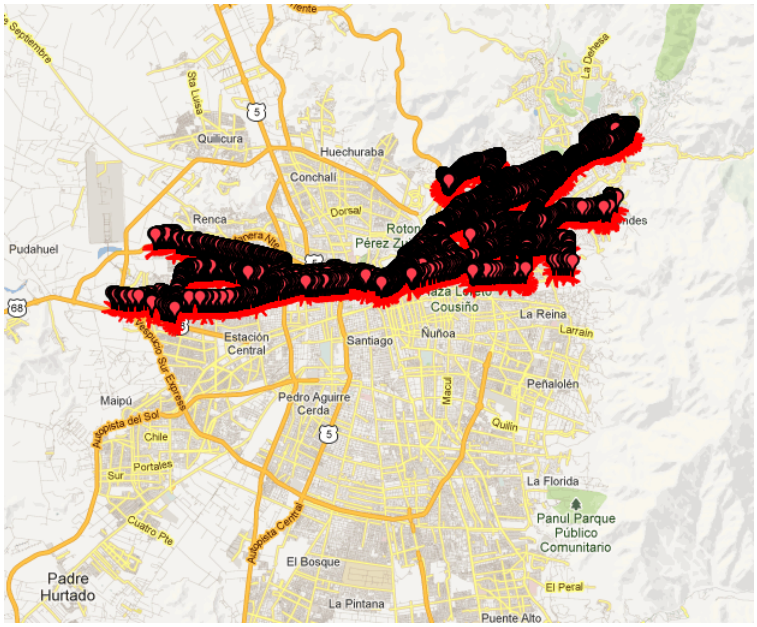


Figura 3.8: Total de arcos que se pueden usar para proyectar tiempos de viaje

El total de recorridos procesados es 11, de los cuales 6 son líneas que cubren el sector oriente de la ciudad y 5 son troncales. Estos últimos conectan distintos sectores de la capital, usando para ello, arterias importantes que definen el mapa urbano. Tanto calles como avenidas relevantes quedan cubiertas por estos recorridos, lo cual permite posteriormente, estimar tiempos y velocidades de viaje.

Los 11 recorridos proyectados utilizan en su conjunto más de 5380 arcos, repartidos en más de 5 comunas, dentro de los cuales, la mayoría son avenidas. De hecho, aproximadamente un 76 % del total de arcos, son avenidas, mientras que solo un 20 % corresponde a calles. Gracias a esto, a pesar de que no se está incluyendo gran parte del grafo en el análisis, los principales circuitos quedan dentro. En la Figura 3.8 están graficados todos los arcos que se han identificado a algún recorrido. Avenidas importantes, tales como Alameda BO,

Providencia, Américo Vespucio, etc. quedan incluidas, lo que permite estimar tiempos de viaje para gran parte del grafo, tan solo utilizando estos 11 recorridos.

No obstante, muchos recorridos presentaron problemas. Arcos inexistentes, desviados o corridos fueron los principales problemas encontrados. Debido a esto, fue necesario arreglarlos para completar el camino. Se detectaron 43 errores, siendo común encontrar 2 o 3 problemas por línea. La mayor cantidad de problemas se encontraron en las calles, llegando a 34, mientras que las avenidas mostraron estar en mejores condiciones mostrando solo 7.

Con los caminos seleccionados y cargados a la tabla `arco_rec` (donde cada `id` de arco se asocia a un `codigo` de recorrido), se comienza con la proyección de los puntos correspondientes a ese recorrido.

Figura 3.7: Ejemplo de puntos GPS que definen un recorrido particular. Izquierda puntos GPS, derecha recorrido que forman estos puntos

Capítulo 4

Proyección de tiempos y velocidades de viaje

El cálculo de tiempos y velocidades de viaje para los buses del sistema de transporte urbano, es un gran desafío, principalmente por la cantidad de información y el grado de desagregación de la data. Este sistema de transporte fue implementado años atrás e incorporó múltiples mejoras, como registro de pago y seguimiento GPS de los buses.

Cada uno de los 6000 buses genera un dato cada 30 segundos, el cual contiene la latitud y longitud en que se encuentra el bus en ese momento, la fecha, horas, minutos y segundos a los cuales se registró el dato, y la patente, recorrido y sentido del bus. Finalmente, se incluye la velocidad instantánea registrada en ese momento, la cual no se tomará en cuenta en este estudio, en parte, debido a que es muy variable en condiciones urbanas. Los buses están constantemente acelerando y desacelerando, y los resultados que se pueden sacar de estos datos no son confiables según los operadores del sistema.

Además, muchos autores han comparado los resultados obtenidos estimando tiempos de viaje usando puntos GPS en vehículos, frente a enfoques alternativos, y la mayoría de ellos avala la utilización de GPS. Por ejemplo, se han utilizado los registros de contadores de tráfico para estimaciones de tiempos y velocidades frente a mediciones de puntos GPS en vehículos de prueba, y se ha concluido que los puntos GPS entregan resultados más precisos [6]. Por consiguiente, el uso de esta fuente de datos se encuentra validada, por lo que se hará uso de ella para estimar.

La velocidad y el tiempo de viaje que se considerarán son aquellos que tienen los vehículos que atraviesan calles, avenidas, autopistas, etc, tomando en cuenta todas las variables que pueden afectar el desplazamiento. Dentro de estos factores, se tiene por ejemplo, semáforos, señales de tránsito y paraderos (importantes en el caso de los buses del Transantiago).

Para obtener valores confiables es crucial el manejo apropiado, tanto de procesamiento como de estimación de los inputs. *Cortés et al.* [6], quienes trabajaron con los mismos datos con los que se cuenta para este proyecto, proponen una metodología flexible y representativa que permite manejar distintos niveles de agregación en el espacio y tiempo. Definen un

diagrama de ejes tiempo-espacio con grillas para cada ruta, donde cada elemento de la grilla está definido por un rectángulo dentro del diagrama. La grilla permite interpolar los puntos tanto espacial como temporalmente, con lo cual los segmentos definidos se asocian a datos precisos. En otras palabras, equivale a que los puntos a proyectar "cayeran" exactamente en los extremos de cada segmento.

En un eje se tiene el tiempo y en el otro la distancia o espacio. Todas las grillas tienen las mismas dimensiones, lo que significa que el horizonte tanto temporal como espacial, están divididos en las mismas proporciones. Un ejemplo de estas grillas se observa en la Figura 4.1. *Cortés et al.* necesitan el momento y posición en que cada bus entra y sale las correspondientes grillas para así calcular la velocidad promedio de este segmento. Para ello, interpolan los puntos, tal y como se muestra en la Figura 4.1 con estrellas.

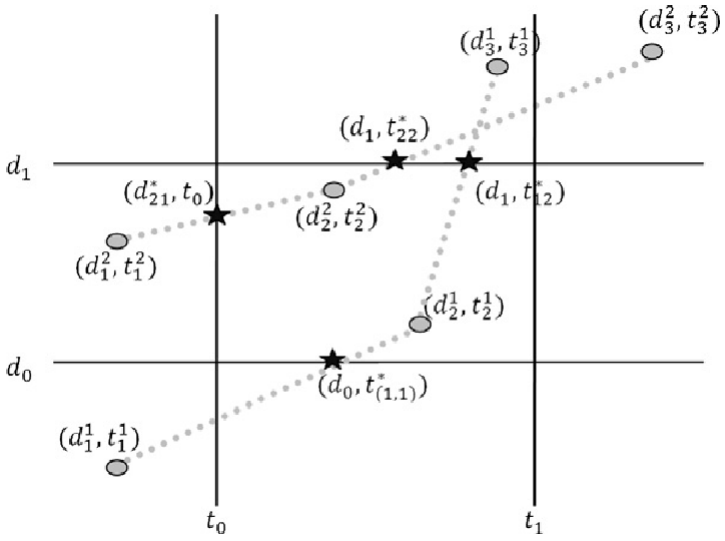


Figura 4.1: Ejemplo de grillas que se utilizan para calcular velocidades de viaje

4.0.1. Modelo propuesto

Se propone un modelo inspirado en el trabajo de *Cortés et al.*, el cual permite proyectar un set de puntos GPS emitidos por un bus. Como se revisó en el Capítulo 3, los recorridos ya están identificados en el mapa, por lo que ya se conoce con exactitud dónde deben ser proyectados los puntos. *Cortés et al.* usan segmentos de tamaño fijo, por ejemplo, de 500 metros. Cuando se tienen los arcos, no es necesario fijar estos segmentos, pues cada uno de ellos es un arco.

Usando este método se deben suponer dos cosas, las cuales están implícitas en el trabajo de *Cortés et al.*:

- La velocidad entre la proyección de un punto GPS g_{i}, y el siguiente $g_{\mathrm{i+1}}$, es constante. En otras palabras, la secuencia de arcos que recorre el bus que generó los puntos GPS, lo hace a velocidad constante entre dos puntos GPS, sin importar la naturaleza de estos arcos.

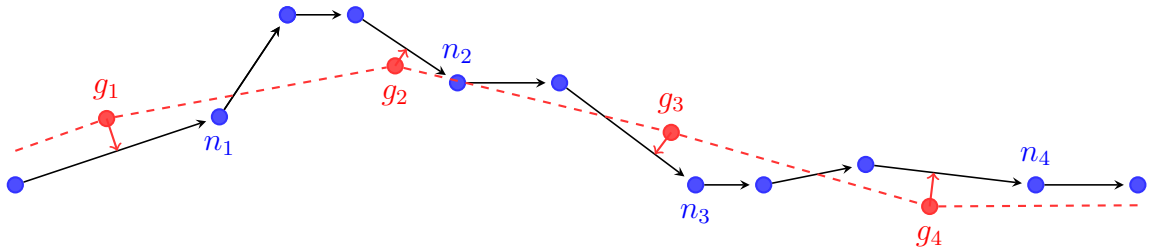


Figura 4.2: Ejemplo puntos gps proyectados sobre un subgrafo para calcular velocidades

- La velocidad con que se avanza dentro de un arco, es constante. Un punto contenido en la linea de nodo inicio y nodo fin que define un arco, se puede desplazar a lo largo de esta linea, modificando el tiempo según la velocidad que tiene ese segmento.

Es común que entre dos arcos, dónde se proyectaron puntos GPS consecutivos que forman un camino, existan varios arcos. Usando este método, es simple calcular la velocidad para estos arcos intermedios.

Un ejemplo de cómo proyectar puntos GPS, se encuentra en la Figura 4.2. Para proyectar simplemente se escoge el arco dentro del camino más cercano al punto que se quiere proyectar y posteriormente, se posiciona esta proyección dentro del arco. Se calcula la velocidad entre un punto g_{i} y el siguiente, $g_{\mathrm{i+1}}$, usando los lugares dónde fueron proyectados. Como se mencionó anteriormente, la velocidad entre estos puntos es constante, la cual además se define como

$$\frac{||p_{\mathrm{i}j^*} - p_{(\mathrm{i+1})j^*}||_{\mathrm{ruta}}}{t_{\mathrm{i},(\mathrm{i+1})}} \tag{4.1}$$

siendo $p_{\mathrm{i}j}$ la proyección del punto i en el arco j, y $t_{\mathrm{i},(\mathrm{i+1})}$ el tiempo entre el punto g_{i} y el $g_{\mathrm{i+1}}$.

Cada proyección, luego, se debe desplazar al nodo fin del arco donde se define, según la interpolación descrita. En este ejemplo el punto g_1 se corre al nodo n_1, el punto g_2, al nodo n_2, y así sucesivamente, utilizando para ello, la velocidad que ya se ha calculado.

Cuando los puntos, g_{i}, tienen un nodo asociado, nuevamente se calcula la velocidad, pero esta vez, usando los nuevos tiempos y ubicaciones. Esta velocidad, se proyecta sobre todos los arcos entre los dos nodos. Por ejemplo, los puntos g_1 y g_2, se asocian a n_1 y n_2, y la velocidad entre estos nodos, la cual se asume constante, se asocia a todos los arcos que están en la secuencia.

4.0.2. Implementación

Comienza este algoritmo de procesamiento, ordenando todos los datos que se usarán según patente y tiempo (fecha y hora), y filtrando por el código y sentido de viaje. De esta manera, los datos quedan dispuestos como viajes, un punto tiene su continuación justo en la fila siguiente. La siguiente consulta SQL permite obtener los datos según estos requisitos, para el recorrido T405:

```
SELECT patente, sentido, TO_CHAR(fechahora,'YYYYMMDD'),
TO_CHAR(fechahora,'SSSS'), latitud, longitud
FROM data
WHERE codigo = 'T405'
ORDER BY patente DESC, fechahora DESC
```

Con los datos ordenados se da paso al procesamiento de ellos. El algoritmo que se usa, es el Algorithm 3. En cada momento se tienen 3 puntos g_{i}, $g_{\mathrm{i+1}}$ y $g_{\mathrm{i+2}}$. El último de ellos, es necesario para desplazar a $g_{\mathrm{i+1}}$, puesto que permite calcular velocidad a la que se corre al siguiente nodo.

Algorithm 3: Algoritmo de proyección para cálculo de velocidad

```
1  p_0 = proyección punto gps inicial
2  p_1 = proyección siguiente punto gps distinto a p_0
3  while (mientras hayan puntos gps) do
4      Sea g siguiente punto proyectado sobre el grafo
5      if g = null o arco de p_1 es igual a arco de g then
6          ir a linea 4
7      if arco de p_0 es igual a arco de p_1 then
8          p_1 = g
9          ir a linea 4
10     if Patente de g es distinta a patente de p_1 o fecha distinta o se proyecto sobre un
       arco que no está después del de p_1 then
11         p_0 = g
12         p_1 = siguiente punto a g
13         ir a linea 4
14     v = velocidad entre p_0 y p_1
15     for cada arco entre p_0 y p_1 do
16         Asignar v a velocidad de ese arco
```

Los distintos casos, ya sea que el punto se proyectó sobre un arco que no tiene sentido, que la patente o la fecha cambia, etc., se detallan en el algoritmo de proyección. Un caso particular es que un punto se proyecte sobre el mismo arco, pues, como se discutió anteriormente, un punto en el mismo arco no cambia los valores de velocidad. La velocidad dentro de un arco es constante.

Una pregunta natural sería el por qué no usar los puntos proyectados sin desplazarlos. La respuesta es que con este método, los datos no se pierden y es posible replicar todo el viaje que hizo un bus. Si, en cambio, se utiliza el punto donde queda en el arco, se debería omitir la velocidad de ese arco, o de alguna forma arbitraria, ponderar las velocidades de un extremo y otro, mecanismo que se considera inapropiado para el estudio.

Se repite este proceso para cada bus de un mismo recorrido, generando una distribución de tiempos de viajes en cada arco. A estos arcos se les asocia datos de tiempos, lo cuales serán distintos para cada bus. Sin embargo, un mismo bus puede originar datos iguales para

un conjunto adyacente de arcos, pues la velocidad que se calcula se asocia a todos los arcos entre dos puntos GPS seguidos.

Capítulo 5

Análisis de distribuciones de tiempos de viaje en arcos y caminos

5.1. Distribución para los tiempos de viaje

El análisis de las distribuciones de probabilidad de los tiempos de viaje y de la velocidad con que se viaja por una ciudad, es un problema ampliamente estudiado. El tiempo de viaje es una de las más importantes variables al momento de medir el desempeño de sistemas de transportes, por lo que diversos estudios han buscado mejorar el desempeño de estos sistemas, intentando disminuir estos tiempos a través de iniciativas para manejar el tráfico.

Sin embargo, el sistema de transporte en una ciudad es complejo y tiene una naturaleza altamente estocástica. Por ejemplo, un accidente o incluso el clima pueden afectar, incrementando los tiempos de desplazamiento desde un punto a otro. En otros escenarios estos tiempos pueden ser menores al promedio, volviendo interesante el análisis de la varianza de estas variables aleatorias y su distribución.

Dada la necesidad de estimar tiempos más confiables en áreas urbanas y así proporcionar un mejor servicio de transporte, muchos estudios han modelado y valorizado la variabilidad de los tiempos de viaje. Se han analizado tanto arterias urbanas como autopistas usando distribuciones paramétricas continuas. Ello ha dado origen a intervalos de confianza a los tiempos que se estiman, los cuales comúnmente suponen distribución normal [10]. Se calculan medias y varianzas, y luego, con criterios de confiabilidad, se construyen dominios asumiendo distribución normal. Así por ejemplo, cuando se estudia el tiempo de viaje por una avenida, o cuando se buscan variables que puedan afectar a estos tiempos, se reportan estadísticas asumiendo que estos tiempos están normalmente distribuidos. Estudios donde se realizan regresiones lineales, estimación de parámetros, etc., comúnmente asumen esta distribución [10].

Sin embargo, la distribución Normal no tiene un ajuste muy preciso de los datos, debido principalmente a la asimetría que presenta la distribución de los tiempos de viaje. *Wardrop*, ya había evidenciado este comportamiento en su estudio en las calles de Londres [31] publicado

en 1952, donde establece que la normal deja de lado este factor que es muy importante incluir.

Herman & Lam [14] también encontraron una importante asimetría de la distribución de estas variables aleatorias, cuando analizaron datos recolectados de las calles de Detroit. Estudiaron varios viajes realizados en automóviles, asumiendo que el tiempo de viaje en diferentes secciones del camino de la red eran independientes entre sí. Además asumieron que el tiempo de viaje en secciones de igual largo eran idénticamente distribuidas. Dentro de sus propuestas para el ajuste de datos, estaba la distribución Gamma y la Lognormal, las cuales lograban explicar la asimetría que evidenciaban estos tiempos.

Posteriormente, *Polus* [22] encontró que la distribución Gamma explicaba muy bien los tiempos de viaje en datos recolectados en Chicago, aunque recientemente, el año 2007, *Faouzi & Maurin* [12], probaron el comportamiento de la Gamma, Normal y Lognormal, concluyendo que la última de éstas permitía explicar mejor los tiempos de viaje. Usaron para ello, dos test estadísticos de ajuste en datos que fueron obtenidos en una autopista de 18 km. de largo. Estos fueron medidos a través de pórticos de pago o electronic toll collection system.

Otros autores que también concluyeron que la Lognormal era apropiada para estas variables aleatorias fueron *Taylor & Richardson* [23]. Ellos estudiaron los tiempos de viaje en arterias urbanas usando autos particulares conducidos en Melbourne, Australia. Dentro de sus resultados, también rechazan la Normal como ajuste.

Finalmente, *Susilawati et al.* [28] estudiaron la distribución Burr, conocida también como generalización de la log logistic. En su investigación utilizaron dos caminos de viaje, los cuales dividieron en links de longitud mínima de 150 m. y máxima de 4000 m. Sus mediciones, al igual que este estudio, fueron recogidas usando tecnología GPS.

Estimaron los parámetros de esta distribución usando máxima verosimilitud, y la compararon con la distribución Normal, Lognormal, Weibull, Gamma y Generalised Pareto y concluyeron que la distribución Burr es la mejor, usando para el contraste el test de ajuste *Kolmogorov-Smirnov*.

En base a los resultados de estos autores, se presentarán dos distribuciones, la Lognormal y la Burr, puesto que son dos de las más aceptadas como ajustes para los tiempos de viaje urbanos, y se presentarán argumentos para utilizar una en vez de la otra.

La distribución Lognormal se plantea en diferentes áreas como por ejemplo, finanzas, telecomunicaciones, economía, etc., en parte, gracias a que es continua y positiva. En economía, por ejemplo, se usa para modelar el ingreso de la población utilizando. Además, en muchas situaciones su uso se justifica gracias a propiedades favorables que presenta su uso, muchas de las cuales hereda de la Normal [7].

Si X es una variable aleatoria Normal de media μ y desviación estándar σ, entonces $Y = \exp(X)$ es una Lognormal, para la cual su función de distribución es

$$f(x;\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{(\ln x-\mu)^2}{2\sigma^2}} \qquad \text{con } x > 0 \tag{5.1}$$

Además, se tiene que la función de distribución acumulada (FDA) se puede expresar usando la función de una Normal estándar.

$$F(x;\mu,\sigma) = \Phi\left(\frac{\ln x - \mu}{\sigma}\right) \qquad \text{con } x > 0 \tag{5.2}$$

La esperanza y varianza de esta variable no es tan directa como la Normal, pero presenta una forma simple para calcularla. En efecto,

$$\mathbb{E}(X) = \mathrm{e}^{\mu+\frac{1}{2}\sigma^2} \tag{5.3}$$

$$\mathbb{V}ar(X) = (\mathrm{e}^{\sigma^2} - 1)\mathrm{e}^{2\mu+\sigma^2} \tag{5.4}$$

Los estimadores tanto de μ como de σ^2 máximos verosímil, no varían significativamente con respecto a la Normal

$$\hat{\mu} = \frac{\sum_k \ln x_k}{n} \tag{5.5}$$

$$\hat{\sigma^2} = \frac{\sum_k (\ln x_k - \hat{\mu})^2}{n} \tag{5.6}$$

Este último estimador se debe recodar que es sesgado, por lo que comúnmente, se reemplaza el denominador n por $n-1$.

La distribución Burr es también bastante conocida actualmente, donde ha encontrado un espacio modelando procesos de tiempo de vida de productos comerciales, tales como seguros. Fue desarrollada por *Burr* en 1942 [5] con el propósito de ajustar una FDA a una gran diversidad de formas de curvas de datos.

En su versión más básica, esta distribución cuenta con dos parámetros, k y c, ambos positivos. Sin embargo, por lo general se agrega un parámetro extra m que permite escalar la función, el cual es mayor a cero también. Esta distribución tiene la siguiente función de distribución

$$f(x;m,c,k) = \frac{ck(x/m)^{c-1}}{m(1+(x/m)^c)^{k+1}} \tag{5.7}$$

y su FDA

$$F(x;m,c,k) = 1 - (1+(x/m)^c)^{-k} \tag{5.8}$$

La distribución Burr tiene una curva flexible y buenas propiedades algebraicas. Tiene una cola que le permite modelar fallas poco frecuentes, y además, su FDA tiene una expresión cerrada, lo que permite calcular fácilmente percentiles. Por último, su flexibilidad permite aproximarla a una gran variedad de distribuciones tales como Log-logistic, Weibull y Extreme value.

No obstante, existen varias razones que permiten apoyar la decisión hacia la Lognormal:

- Tiene menos parámetros: En efecto, para la Lognormal solo se tienen que estimar dos parámetros, μ y σ, mientras que en el caso de la distribución Burr, son tres, m, c y k.
- Estimación sencilla: Los parámetros de la Lognormal pueden ser estimados usando simples ecuaciones. En cambio, los tres parámetros de Burr se deben estimar usando máxima verosimilitud, ya sea utilizando el método de quasi-Newton, gradientes conjugados, el método de simulated-annealing, etc. Estos métodos generalmente requieren valores iniciales donde partir y no siempre convergen.
- Mecanismos estudiados para la suma de variables: La Lognormal tiene varios métodos que se han estudiado para determinar la distribución de la suma de variables aleatorias. Existen mecanismos sencillos por igualación de momentos, cálculo de esperanzas y métodos usando funciones generadoras de momentos, entre tantos otros. Por otra parte, la suma de distribuciones Burr, no ha sido tan estudiada. Existen métodos que permiten sumar variables pero que asumen i.i.d. (*Kortchak & Albrecher* [16]), supuesto muy restrictivo para el caso de tiempos de viaje en una red de transporte. Además, el cálculo es engorroso y requiere convergencias de series.

Finalmente, se decide utilizar la distribución Lognormal respaldado por estos argumentos y porque cuenta con el apoyo de varios autores para su uso en esta materia.

5.2. Estimación de parámetros y suma de variables

Muchos estudios analizan conjuntamente el tiempo de viaje y la velocidad de desplazamiento por una ciudad, lo cual no presenta problemas cuando se pretende estudiar el comportamiento de la suma de variables. En efecto, los tiempos de viaje pueden ser sumados, mientras que la velocidad, al menos de forma directa, no. Es por esto que la variable que se utilizará en el estudio de arcos de Santiago, es el tiempo de viaje a través de un segmento, debido a que lo que se busca replicar es el comportamiento de un camino.

Si bien algunos autores han publicado resultados que apuntan que los tiempos de viaje son independientes entre segmentos, como por ejemplo *Taylor & Richardson* [23], otros han medido estas correlaciones y han concluido que se deben considerar [12].

Dentro de los puntos a estudiar será si estas correlaciones son importantes o no. El factor de correlación entre dos Lognormales, X e Y, se estima como

$$\rho = \frac{\sum_{\mathrm{i}}^{n}(\ln x_{\mathrm{i}} - \hat{\mu_X})(\ln y_{\mathrm{i}} - \hat{\mu_Y})}{\sqrt{\sum_{\mathrm{i}}^{n}(\ln x_{\mathrm{i}} - \hat{\mu_X})^2 \sum_{\mathrm{i}}^{n}(\ln y_{\mathrm{i}} - \hat{\mu_Y})^2}} \tag{5.9}$$

Estos factores se usarán, en el caso de que sean estadísticamente distintos de cero, para sumar tiempos de viaje a lo largo de un camino de varios arcos. Los resultados asociados a estas correlaciones, si son significativos o no, importancia relativa, etc., se presentan en los resultados.

Cada arco del mapa tiene una distribución para el tiempo de viaje, el cual se explica estimando los parámetros de la Lognormal respectiva, los que se relacionan a través de los factores de correlación, siempre y cuando estos sean significativos y sea apropiado incluirlos.

Sean Y_k variables aleatorias lognormalmente distribuidas, de parámetros distintos y $X_k \sim N(\mu_{X_k}, \sigma^2_{X_k})$ tal que $\exp(Y_k) = X_k$. Además, sea $Y = \sum_{k=1}^{K} Y_k$, la suma de estas variables. La pregunta es, ¿cómo se puede representar esta variable aleatoria?

Uno de los primeros puntos que se deben tener presente es que no se conoce una forma cerrada para representar esta suma, lo que se traduce en que no se pueden establecer ecuaciones simples para estimar los parámetros de la distribución resultante. Visto de otra forma, la integral que definen esta suma, no es posible calcularla, ni tampoco representarla haciendo uso de la FDA de la Normal estándar. Afortunadamente, el estudio de esta suma está bastante investigado.

Los métodos propuestos en la literatura se pueden gruesamente dividir en dos categorías. Tanto los métodos de *Fenton & Wilkinson* (FW) [13], *Schwartz & Yeh* (SY) [25] y *Beaulieu & Xie* (BX) [4] aproximan la suma de Lognormales a una sola variable aleatoria Lognormal, es decir, estos tres métodos asumen que la distribución resultante es Lognormal. Empíricamente se puede notar que esta suma sigue esa distribución, idea que apoya el supuesto de estos métodos. Por otra parte, los métodos de *Farley* [27], *Slimane* [26] y *Schleher* [24] calculan una composición de distribuciones basados en las propiedades de la Lognormal, en otras palabras, establecen que el resultado es una composición de varias Lognormales por tramos. Esta composición se puede especificar de varias formas. Por ejemplo, el método de *Schleher* particiona el rango de la suma de Lognormales en tres segmentos, cada uno de ellos aproximando a una Lognormal distinta.

El método de *Farley*, BX y el de *Slimane* suponen independencia en las variables que se sumarán y no tienen extensiones para la suma variables correlacionadas, un factor que como se verá más adelante es relevante. Debido a que no se quiere dejar fuera esa opción, no se estudiarán en este trabajo.

En el estudio de *Abu-Dayya & Beaulieu* [2], se compara el método de FW, SY y Schleher, para variables dependientes. En el contexto de la interferencia de las señales de sistemas móviles (contexto en el que se desarrolla el trabajo de estos autores), estas superposiciones aleatorias podían presentar correlaciones entre unas y otras. Para calcular la probabilidad de interrupción, se debe tomar la suma de las interferencias. *Abu-Dayya & Beaulieu* estudian estos métodos para sumar variables de esta naturaleza y concluyen que el método de FW es el mejor para explicar los fenómenos en sistemas de comunicación móvil. Es decir, de los tres métodos, *Abu-Dayya & Beaulieu* establecen que FW es el que encuentra la mejor distribución para la suma de variables.

Además de el apoyo de *Abu-Dayya & Beaulieu*, el método de FW es un método validado por varios autores para explicar la suma de variables aleatorias Lognormales. Cuenta también con otras ventajas, tales como una extensión sencilla para el caso de variables correlacionadas y una simple manera de calcular parámetros. De esta forma, esta distribución se vuelve una alternativa interesante para estudiar los tiempos de viaje.

El método de FW supone X_{i}, i $= 1, ..., n$ v.a. Normales con $\sigma_{\mathrm{i}j}$ la correlación entre $\exp(Y_{\mathrm{i}}) = X_{\mathrm{i}}$ y $\exp(Y_j) = X_j$. Además, se cumple que $\sigma_{\mathrm{ii}} = \sigma^2_{X_{\mathrm{i}}}$. Para determinar la media y varianza de la Lognormal que aproxima la $\sum_{\mathrm{i}} y_{\mathrm{i}}$, se resuelve el siguiente sistema:

$$\mu_Y = \ln\left(\sum_{k=1}^{K} \mathrm{e}^{\mu_{X_k} + \frac{\sigma^2_{X_k}}{2}}\right) - \frac{\sigma^2_Y}{2} \tag{5.10}$$

$$\sigma^2_Y = \ln\left(\frac{\sum_{k=1}^{K}\sum_{j=1}^{K} \mathrm{e}^{\mu_{X_k}+\mu_{X_j}+\frac{1}{2}\left(\sigma_{X_{\mathrm{i}}}+\sigma_{X_j}\right)}(\mathrm{e}^{\sigma_{\mathrm{i}j}}-1)}{\left(\sum_{k=1}^{K} \mathrm{e}^{\mu_{X_k}+\sigma^2_{X_k}/2}\right)^2} + 1\right) \tag{5.11}$$

Dos ecuaciones que permiten estimar de forma muy simple la suma de Lognormales correlacionadas, encontrando los parámetros μ_Y y σ^2_Y equivalentes.

Sin embargo, *Mehta et al.* [18] mencionan los problemas que presenta este método, siendo incapaz de predecir de buena manera la distribución de la head portion, es decir, pequeños valores de la distribución o la parte de la distribución más cercana al cero. Sin embargo, reconoce que es muy preciso cuando se trata de estimar la tail portion, osea, valores mayores. Su estudio, compara este enfoque y otros más con uno que proponen ellos. En su propuesta hacen uso de la función generadora de momentos (FGM), estableciendo que con este método se pueden sortear los problemas que tienen los otros.

Mehta et al. inicialmente recurren a la FGM de Y, variable aleatoria, la cual se define como

$$\Psi_Y(s) = \int_0^\infty \exp(-sy) p_Y(y) \mathrm{d}y \tag{5.12}$$

donde $p_Y(x)$ es la función de distribución.

Si se asume la variable y está Lognormalmente distribuida, desarrollando la integral y además usando la serie de *Gauss-Hermite* para calcular la función de la ecuación resultante, se tiene la siguiente igualdad

$$\begin{aligned}
\Psi_Y(s, \mu_X, \sigma_X) &= \int_0^\infty \exp(-sy) \frac{1}{y\sigma_X\sqrt{2\pi}} \exp\left(-\frac{(\ln y - \mu_X)^2}{2\sigma^2_X}\right) \mathrm{d}y \\
&= \int_{-\infty}^\infty \frac{1}{\pi} \exp\left[-s\exp(\sqrt{2}\sigma_X z + \mu_X)\right] \exp(-z^2)\mathrm{d}z \\
&= \sum_{n=1}^{N} \frac{w_n}{\sqrt{\pi}} \exp\left[-s\exp(\sqrt{2}\sigma_X a_n + \mu_X)\right] + R_N
\end{aligned} \tag{5.13}$$

donde μ_X y σ_X son la media y la desviación estándar de la normal $X = \ln(Y)$. N es el orden de integración de Hermite y R_N es un residuo que es decreciente en la medida que aumenta N. Los pesos w_n y a_n son términos que están tabulados en [1]. *Mehta et al.* proponen que $N = 12$ es suficiente para conseguir una buena aproximación de cálculo, volviendo prescindible el residuo.

Para el caso general de K variables aleatorias correlacionadas Lognormales, $\{Y_k\}_{k=1}^K$, con sus correspondientes variables normales $\{X_k\}_{k=1}^K$, las cuales tienen una matriz de correlaciones $\mathbf{C}$ arbitraria. Las variables $X_k = \ln(Y_k)$ tienen una distribución conjunta dada por

$$p_X(\mathbf{x}) = \frac{1}{(2\pi)^{K/2}|\mathbf{C}|^{1/2}} \exp\left(-\frac{(\mathbf{x}-\mu)'\mathbf{C}^{-1}(\mathbf{x}-\mu)}{2}\right) \tag{5.14}$$

donde μ es el vector de medias de las variables X_k.

Entonces, la FGM de la variable $Y = Y_1 + ... + Y_k$ se puede escribir como [18]

$$\Psi_{(\sum_{k=1}^K Y_k)}(s) = \int_{-\infty}^{\infty} \frac{1}{(2\pi)^{K/2}|\mathbf{C}|^{1/2}} \prod_{k=1}^{K} \exp(-s\exp(x_k)) \exp\left(-\frac{(\mathbf{x}-\mu)'\mathbf{C}^{-1}(\mathbf{x}-\mu)}{2}\right) \mathrm{d}\mathbf{x} \tag{5.15}$$

Sea $\mathbf{C_{sq}}$ la raíz cuadrada de la matriz de correlación $\mathbf{C}$, es decir, $\mathbf{C} = \mathbf{C_{sq}}\mathbf{C'_{sq}}$. Se puede usar la siguiente transformación para simplificar el cálculo

$$\mathbf{x} = \sqrt{2}\mathbf{C_{sq}}\mathbf{z} + \mu \tag{5.16}$$

o equivalentemente

$$x_k = \sqrt{2}\sum_{j=1}^{K} c'_{kj} z_j + \mu_k \qquad \text{para } k = 1, ..., K \tag{5.17}$$

Aquí c'_{kj} es el $(k,j)^{th}$ elemento de C_{sq}. Por lo tanto, la FGM queda como

$$\begin{aligned}
\Psi_{(\sum_{k=1}^K Y_k)}(s,\mu,\mathbf{C}) &= \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} \frac{1}{\pi^{K/2}} \prod_{k=1}^{K} \exp\left(-s\left[\exp\left(\sqrt{2}\sum_{j=1}^{K} c'_{kj}z_j + \mu_k\right)\right]\right) \\
&\qquad \exp(-\mathbf{z}'\mathbf{z})\mathrm{d}z_1\mathrm{d}z_2\cdots\mathrm{d}z_K \\
&= \sum_{n_K=1}^{N}\cdots\sum_{n_1=1}^{N} \frac{w_{n_1}\cdots w_{n_K}}{\pi^{K/2}} \prod_{k=1}^{K} \exp\left(-s\exp\left(\sqrt{2}\sum_{l=1}^{K} c'_{kl}a_{n_l} + \mu_k\right)\right) + R_N^{(K)} \\
&= \sum_{n_K=1}^{N}\cdots\sum_{n_1=1}^{N} \left[\prod_{k=1}^{K} \frac{w_{n_k}}{\sqrt{\pi}}\right] \exp\left(-s\sum_{k=1}^{K}\left[\exp\left(\sqrt{2}\sum_{j=1}^{K} c'_{kj}a_{n_j} + \mu_k\right)\right]\right) + R_N^{(K)}
\end{aligned} \tag{5.18}$$

donde $R_N^{(K)}$ es un residuo resultante de las K integrales que se deben calcular.

Finalmente, se debe resolver un sistema de ecuaciones no lineales, donde μ_X y σ_X son parámetros que se busca estimar, y s es una variable que permite flexibilizar el criterio para estimar. Por consiguiente, $Y = \exp(X)$, tiene parámetros que se pueden estimar resolviendo

$$\Psi_Y(s_\mathrm{i}, \mu_X, \sigma_X) = \Psi_{(\sum_{k=1}^K Y_k)}(s_\mathrm{i}, \mu, \mathbf{C}) \qquad \text{para i} = 1, 2 \tag{5.19}$$

Son dos ecuaciones puesto que hay dos variables que se deben encontrar.

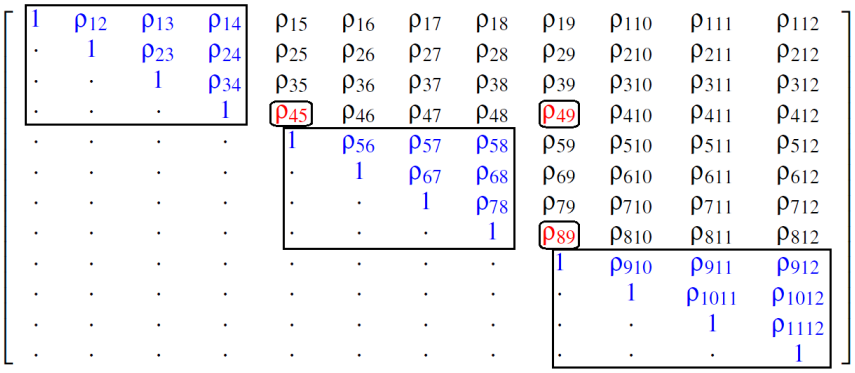


Figura 5.1: Matriz de correlaciones para 12 arcos consecutivos en un camino

En [18] se mencionan las múltiples ventajas que tiene este método con respecto a otros como el de FW, SY y el de BX, mencionando que se tiene la flexibilidad para ajustar distintas porciones de la curva según se requiera, que permite trabajar con Lognormales correlacionadas y que es muy preciso incluso para un amplio espectro de medias y varianzas.

Sin embargo, presenta problemas para usarse directamente para este estudio. En el mejor de los casos, un camino de 1400 metros debería contar con aproximadamente 25 arcos, cada uno con su media, varianza y posibles correlaciones con cada uno de los otros arcos. La ecuación 5.18 requiere en este caso de 25 sumatorias y debe sumar 25^{12} términos, cada uno tan pequeño que es difícil representarlo. Como consecuencia, es computacionalmente muy complicado, seguir el método tal y como se presenta. Se debe simplificar de alguna forma este cálculo.

Se identificó que con hasta 4 suma de lognormales, el método de Metha encontraba resultados en un tiempo razonable. Usando grupos de 4 se juntan los arcos e iterativamente se disminuye el total hasta alcanzar una sola variable aleatoria. Debido a que se debe preservar la correlación entre los arcos que no alcanzan a agruparse se mantienen las correlaciones entre los arcos adyacentes a cada grupo.

Por ejemplo, sea $N = \{a_1, a_2, ..., a_n\}$ un conjunto de arcos consecutivos con una matriz simétrica C con coeficientes c_{ij} con i $\in \{1, ..., n\}$ y $j \in \{1, ..., n\}$. Estos arcos se agruparían en grupos de 4, usando el método de Mehta. De aquí queda un nuevo arco a'_i como resultado de agrupar $a_{4i-3}, ..., a_{4i}$; a'_{i+1} como resultado de agrupar $a_{4i+1}, ..., a_{4i+4}$; así para los n arcos. La correlación para a'_i y a'_{i+1} será igual a $c_{4i,4i+1}$, mientras que la que tenga a'_i y a'_{i+2}, será $c_{4i,4i+5}$.

En la Figura 5.1 se muestra un ejemplo para 12 arcos de un camino. Con azul se representan los grupos que se calculan usando el método de Metha, y con rojo las correlaciones que se mantienen una vez que ya se tienen los parámetros equivalentes para cada segmento. Por ejemplo, el valor ρ_{45} pasa a ser la correlación entre el resultado del primer grupo y el del segundo. Asimismo, ρ_{49} del primero con el tercer grupo, y ρ_{89} de este último con el segundo.

Cuando ya se han agrupado todos los arcos y se ha conformado la matriz de correlaciones equivalente, se repite el proceso, lo cual termina finalmente cuando se tiene un solo arco. A

esta versión alterada de Mehta, se le llamará Mehta modificado o Mehta*.

En este trabajo se evaluarán los distintos métodos de FW, Mehta y Mehta modificado para la distribución de una suma de tiempos de viaje en un camino del grafo. Se reportarán los resultados de ajuste para cada uno de ellos.

5.3. Criterios de ajuste

Cuando se busca estudiar la distribución de probabilidad de una variable aleatoria, lo común es asumir una distribución en particular para esta variable, y luego realizar un test de χ^2 de bondad de ajuste. El primer inconveniente es que la distribución de los tiempos es continua, y para usar este test se debe particionar de forma arbitraria el intervalo. Qué tan bien se ha particionado este intervalo no es claro y además no siempre tienen relación con la naturaleza de las variables ([7]), por lo tanto, se requiere formular un test alternativo.

A continuación se presenta un test no paramétrico para evaluar ajustes para distribuciones continuas, el test de Kolmogorov-Smirnov [7], el cual es usualmente utilizado en este tipo de estudios [12].

Como las variables son continuas, y aprovechando que con probabilidad 0, dos valores son iguales, se definirá la función $F_n(x)$, la cual se construye a partir de los valores sampleados de $x_1, ..., x_n$ de velocidad. A esta función se le llama *sample distribution function*, sample (CDF) o función de distribución empírica (FDE). Al usar esta función se está asumiendo que todo los valores son distintos.

$$F_n(x) = \frac{1}{n} \sum_{\mathrm{i}=1}^{n} I_{X_\mathrm{i} \leq x} \tag{5.20}$$

Aquí $I_{X_\mathrm{i} \leq x}$ toma el valor de 1 si $X_\mathrm{i} \leq x$, y el valor de 0 si no.

Por ejemplo, sean $\{x_\mathrm{i}\}_{\mathrm{i}=1}^{n}$ valores observados de $\{X_\mathrm{i}\}_{\mathrm{i}=1}^{n}$. Para cada número $-\infty < x < \infty$, se define $F_n(x)$ como la proporción de números que son menores o iguales a x. Si estos fuesen k, entonces $F_n(x) = k/n$, lo cual implícitamente asigna probabilidad $1/n$ a cada punto.

Además, sea $F(x)$ la función de distribución de la cual se supone que se samplea $x_1, ..., x_n$. Por la ley de los grandes números se tiene la siguiente convergencia

$$F_n(x) \xrightarrow[n\to\infty]{p} F(x) \qquad \text{para } -\infty < x < \infty \tag{5.21}$$

La convergencia en probabilidad de una v.a. X_n, descrito como $X_n \xrightarrow[n\to\infty]{p} c$, significa que en la medida que aumenta la muestra de tamaño n, el valor de X_n se acerca a c con una probabilidad cada vez más cercana a 1.

Esta relación revela que para cada punto x, el valor de la FDE, $F_n(x)$, va a converger al número que define $F(x)$ de la variable aleatoria de donde fue sampleado. Ejemplo de esta función se encuentra en la Figura 5.2.

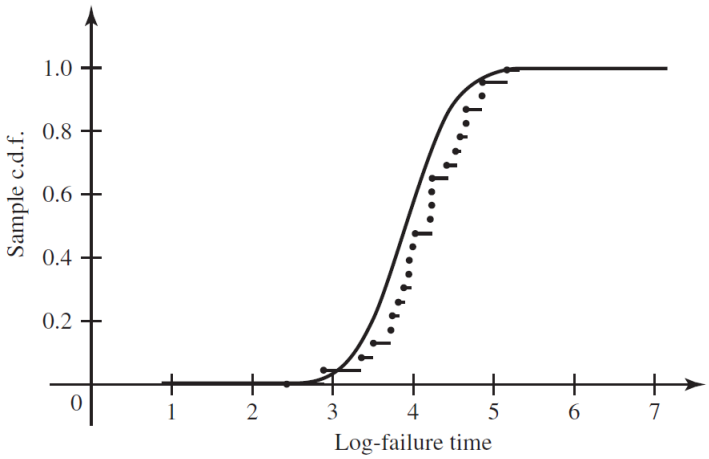


Figura 5.2: Ejemplo de distribución de distribución empírica

Un resultado que se explica en [7], es el lemma de Glivenko-Cantelli, en el cual se establece que $F_n(x)$ converge uniformemente a $F(x)$:

Sea F_n la función de distribución empírica de un sample $X_1, ..., X_n$, de una distribución i.i.d. Se define

$$D_n = \sup_{-\infty<x<\infty} |F_n(x) - F(x)| \tag{5.22}$$

Entonces, $D_n \xrightarrow[n\to\infty]{p} 0$.

El valor que toma D_n para una realización de $X_1, ..., X_n$ es aleatorio, el cual en la medida que n es mayor D_n será menor, y por lo tanto, $F_n(x)$ estará más cerca de $F(x)$. Por lo tanto, en caso de que $F(x)$ fuese desconocida, FDE podría utilizarse como estimador de esta función.

El test de Kolmogorov-Smirnov para una sola hipótesis plantea que una distribución desconocida $F(x)$ es una función continua acumulada, $F(x)^*$, en su hipótesis nula. En su hipótesis alternativa se plantea que esta $F(x)$ es una distribución distinta.

$$\begin{aligned} H_0: &\quad F(x) = F(x)^* \qquad \text{para} -\infty < x < \infty \\ H_1: &\quad \text{La hipótesis nula no es cierta} \end{aligned} \tag{5.23}$$

Este test es no paramétrico debido a que la distribución desconocida proviene de un sample aleatorio que puede ser cualquier distribución continua.

Será D_n^* el supremo de la diferencia entre $F_n(x)$ y la función con la cual se está contrastando, $F(x)^*$. Si la hipótesis nula fuese cierta, el valor de D_n^* debería ser un valor pequeño,

mientras que si no lo fuese, entonces debería ser mayor. Luego, es razonable contrastar la H_0 si $\sqrt{n}D_n > c$, con c un valor apropiado.

Si la hipótesis nula es cierta, entonces para cada valor $t > 0$, se cumple que [30]

$$\lim_{n\to\infty} Pr(\sqrt{n}D_n \leq t) = 1 - 2\sum_{\mathrm{i}=1}^{\infty}(-1)^{\mathrm{i}-1}\mathrm{e}^{-2\mathrm{i}^2t^2} \quad (5.24)$$

$$= \frac{\sqrt{2\pi}}{t}\sum_{k=1}^{\infty}\mathrm{e}^{-(2k-1)^2\pi^2/(8t^2)} \quad (5.25)$$

$$= H(t) \quad (5.26)$$

Dentro de las igualdades presentadas, la expresión 5.25 es analizada por *Marsaglia et al.* [30] quienes la presentan como una alternativa a la serie típica. Su versión la recomiendan para valores pequeños de t, y al mismo tiempo señalan que es más estable y toma menos tiempo en calcularse. Esta serie es muy útil puesto que es rápida de calcular y simple de implementar.

La probabilidad recién definida permite contar con un criterio entre 0 y 1 para evaluar ajustes, siendo el primer valor un indicador de que los datos ajustan perfectamente, y el segundo, uno de que lo hace muy mal.

Para adaptar este test a evaluar si una distribución de los tiempos de viaje sigue una cierta distribución Lognormal, se compara con una Lognormal de parámetros $\hat{\mu}$ y $\hat{\sigma}$, los cuales son estimados a partir de los datos.

En la Figura 5.3 se muestran seis ejemplos de ajustes de distribución para una lognormal. De ellos, los primeros tres presentan valores que se consideran aceptados para este estudio, pues son menores a 95 %, límite que se utiliza en diversos estudios para rechazar ([12], [28]). Los últimos tres cuadros se rechazan, siendo (d) prácticamente el límite de este criterio. El primer histograma (a) se le asocia un coeficiente de 0.12 %, un número muy bajo, lo cual se traduce en que este perfil ajusta muy bien. (b) tiene un valor más alto, 25.01 %, lo que sigue siendo bueno pero menos que (a). (c) tiene un 50.06 %, solo aceptable, mientras que (d) es rechazada con su 95 %. (e) y (f) son malos ajustes, ambos rechazados. Sin embargo, (e) es mucho mejor que (f) puesto que este último prácticamente alcanza el 1.

El estadístico D_n se creó con el fin de testear la hipótesis nula $F(x) = F^*(x)$. Con un 95 % comúnmente se rechaza la H_0. De igual manera, se usa este límite, con la diferencia que aquí se usa para dividir aquellos (arcos, hora) que tienen una distribución en particular, y quiénes no. Es así que Kolmogorov-Smirnov se usa como un criterio para seleccionar, y no como un test clásico estadístico. No se busca rechazar simplemente la hipótesis nula, lo que se quiere es contar con un sistema para clasificar distribuciones entre las que se pueden explicar usando la Lognormal, y las que no.

También este estadístico ayuda a comparar dos muestras, sin la necesidad de suponer distribuciones de antemano. El método es el mismo que el que se usa para una sola muestra

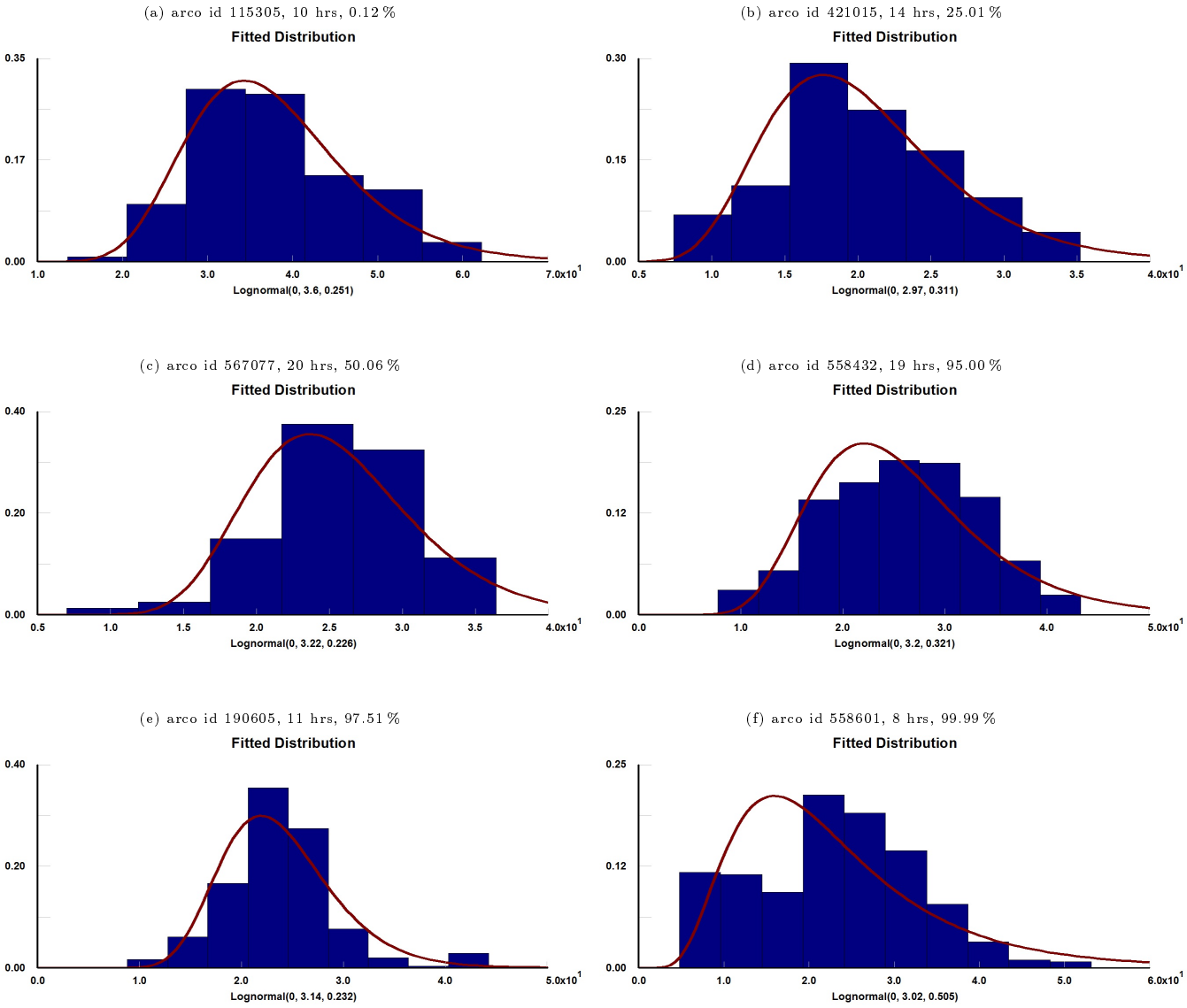


Figura 5.3: Ajustes lognormal para distintos arcos-hora y el valor del estadístico de criterio

que se quiere contrastar, la diferencia es que algunos parámetros cambian. En [7] se menciona como es este test para dos samples. En este caso la hipótesis nula y la alternativa son

$$\begin{aligned} H_0: &\quad F(x) = G(x) \qquad \text{para} -\infty < x < \infty \\ H_1: &\quad \text{La hipótesis nula no es cierta} \end{aligned} \tag{5.27}$$

donde $F(x)$ es una función continua desconocida, al igual que $G(x)$. Una muestra aleatoria de m observaciones, $X_1, ..., X_m$, es tomada desde $F(x)$, la cual es independiente de la muestra de tamaño n proveniente de $G(x)$.

Sea $F_m(x)$ la FDE calculada a partir de $X_1, ..., X_m$ y $G_n(x)$ la función para $Y_1, ..., Y_n$. Usando estas funciones, se calcula el parámetro de distancia máxima entre una y otra,

$$D_{mn} = \sup_{-\infty<x<\infty} |F_m(x) - G_n(x)| \tag{5.28}$$

Cuando la hipótesis nula es cierta y $F(x)$ y $G(x)$ son funciones iguales, las FDE $F_m(x)$ y $G_n(x)$ tienen a los mismos valores. Según el lemma de Glivenko-Cantelli, equivalente al test con una sola muestra, $D_{nm} \xrightarrow[n\to\infty]{p} 0$, cuando $m \overset{\infty}{\to}$ y $n \overset{\infty}{\to}$.

Esta distancia permite de manera equivalente, establecer un criterio de ajuste [7]

$$\lim_{n\to\infty} Pr\left(\left(\frac{mn}{m+n}\right)^{1/2} D_{mn} \leq t\right) = H(t) \tag{5.29}$$

Las condiciones para aceptar o rechazar un ajuste son los mismo que se mencionaron para el test de un solo sample.

Capítulo 6

Resultados

6.1. Distribución y caracterización de los tiempos de viaje por arco

Se procesaron 1036 arcos, y para cada uno de ellos, se tomaron 19 horas, comprendidas entre las 6:00 de la mañana y las 00:59 de la noche. Los datos provienen de dos recorridos no troncales, C11 y C07, ambos cubren parte de la zona oriente de la ciudad, los cuales fueron identificados usando los algoritmos explicados en el capítulo 3. Entre la 1:00 y 5:59 de la madrugada, no transitan los recorridos analizados, por consiguiente, no se tienen datos para esas horas.

Estos recorridos se estudian usando los datos tanto de ida como de vuelta de los buses. Se toman proyecciones desde que un bus parte de un origen a su destino, y luego en el otro sentido. Además, con el objetivo de no incluir datos de días con perfiles distintos, se excluyeron los días sábados y domingos, dejando solo los comprendidos entre lunes y viernes. La exclusión de estos días también se realiza en el trabajo de *Cortes et al.* [6].

El criterio de Kolmogorov-Smirnov se vuelve muy inestable con pocos datos, de ahí que solo se tomaran pares (arco,hora) con al menos 50 datos. Debido a que en la noche, en particular a las 23 y 0 horas, el número de datos no fue suficiente para cubrir este requisito, las horas que tienen mayor peso en los resultados obtenidos, están entre las 7:00 y 22:59 horas.

El intervalo de tiempo analizado que se perjudica más con esta restricción de 50 datos está comprendido entre las 0:00 y las 0:59, intervalo que no contiene datos de ajuste para ninguno de los 1036 arcos. Todos ellos presentan menos de 50 registros, lo que no permite establecer afirmaciones con respecto a los ajustes de distribución en ese horario. Así es como el período estudiado se reduce al incluido entre las 6:00 y 23:59 hrs.

Cabe recordar que los análisis de distribución se hacen separando los tiempos según las horas del día, puesto que el comportamiento que presenta cada horario puede ser muy distinto con respecto a otro. Esta diferencia puede conducir a conclusiones erróneas. Posteriormente

se comenta posible alternativas para esta segmentación.

Arcos	Resultados			Parámetros (%)		Intervalo 95 %	
	$\leq 95\,\%$	excluidos	total	$\hat{p}$	$\hat{\sigma}$	inferior	superior
Todos	9899	0	16524	59.91	0.38	59.14	60.67
Sin ≥ 15	9722	139	14122	68.84	0.39	68.06	69.62
Sin ≥ 13	9555	187	13303	71.83	0.39	71.05	72.61
Sin ≥ 10	8970	289	11621	76.72	0.39	76.41	77.97

Tabla 6.1: Proporción de pares (arco,hora) que se consideran distribuidos lognormal. El total de arcos es de 1036

La Tabla 6.1 contiene la información del número de arcos por hora que cumplen con el criterio establecido para discriminar entre aquellos que tienen una distribución Lognormal, y quienes no según el criterio que se ha presentado. El total de arcos, 1036, generan 16 mil combinaciones cuando se separan en las 18 horas de estudio. La primera columna muestra cuántas de estas combinaciones cumplen con el criterio de ajuste($\leq 95\,\%$). Además, se incluye el ratio y un intervalo de confianza (al 95 % aprox.) para la proporción de pares (arco,hora) que cumplen esta condición, cuando la cantidad de datos es n.

$$\hat{p} \pm 2 \cdot \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \tag{6.1}$$

La primera fila muestra el total de arcos, incluyendo tanto los que tienen muchas de sus horas bien ajustadas, como los que no. De ellos, aproximadamente un 60 % se clasifica correctamente, proporción que aumenta sustancialmente cuando se excluyen aquellos arcos que no ajustan en 15 de las 18 horas de análisis. Aquí se están tomando los ajustes para cada par (arco,hora) pero se están excluyendo aquellos arcos que dentro de las 18 horas de análisis, el coeficiente supera el 95 % en 15 de ellas. Con esto se busca prescindir de aquellos arcos que no aportan, y que presentan problemas en prácticamente todo el día.

Si se continua quitando arcos que no ajustan bien, el ratio sigue aumentando, llegando a aproximadamente un 77 %. Con esto se puede inferir que solo un grupo pequeño de estos arcos no cumplen con la distribución que se está proponiendo, mientras que una importante mayoría sí lo está haciendo.

Esto último queda en evidencia en la Figura 6.1. Este gráfico de barras tiene en su eje horizontal el número de horas que ajusta un arco. El eje vertical muestra el porcentaje del total que tiene ese número de aciertos.

Si se toman arcos con 0, 1, 2 y 3 horas con pésimo ajuste, su peso en el total no supera el 20 %. Por otro lado, los muy buenos, 12, 13, 14 ó más, están muy por encima de los arcos conflictivos. En efecto, la suma de sus contribuciones llegan casi al 40 %. Con esto se quiere establecer que existen dos grupos de arcos en la muestra, aquellos que siguen la distribución Lognormal a lo largo del día, y aquellos que su distribución no ajusta en ningún horario. Éstos últimos no son una gran mayoría del total, y deben presentar factores comunes que los hacen seguir este comportamiento.

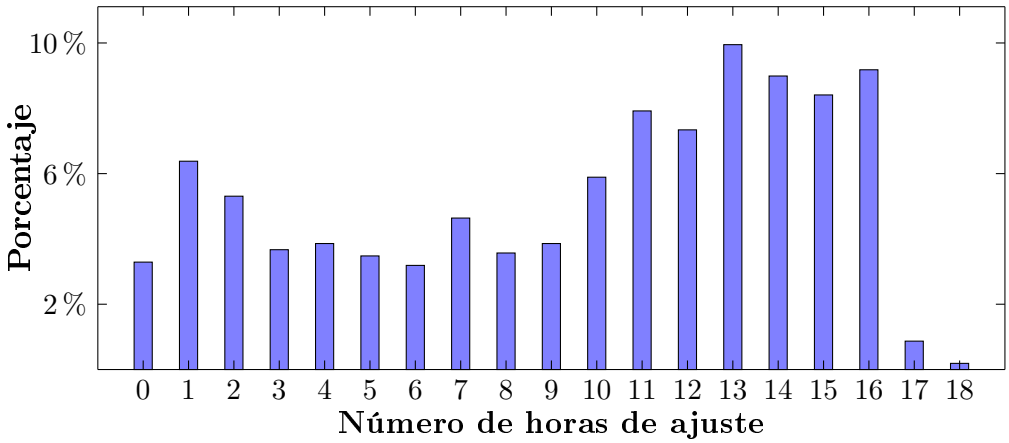


Figura 6.1: Porcentaje de arcos de la muestra estudiada que ajustan bien ($\leq 95\,\%$) una cantidad de veces dada.

6.1.1. Caracterización geográfica

Uno de las metas planteadas es caracterizar tanto los arcos con buenos ajustes como los malos, una vez que ya se ha evidenciado esta diferencia. Se busca identificar el perfil tipo de los unos y de los otros. Por ejemplo, permitir esclarecer si son avenidas, calles, carreteras, etc. los que presentan buenos ajustes; o si los malos arcos están cerca de las intersecciones; cuáles son las peores horas del día para estas distribuciones, etc.

Con este fin, se grafican algunos ejemplos de estos arcos conflictivos. En la Figura 6.2 se grafican segmentos que no ajustan en prácticamente ninguna hora, que no tienen un comportamiento claro en su distribución. En el mapa se señala algunos factores externos que pueden estar influyendo en la distribución, como semáforos y resaltos o lomos de toro.

Se observa como, por ejemplo, en las intersecciones se acumulan varios de estos arcos, presentando un efecto antes de llegar al cruce y una vez que ya se ha pasado. Los semáforos, los cuales se identifican con un rombo naranjo, se distinguen como un factor crítico en las distribuciones. Es posible encontrarlos en gran parte de las zonas con problemas. Incluso, en el cuadro de abajo a la derecha de la Figura 6.2, se observa el efecto que este tiene en un segmento completo, siendo el único factor presente que podría estar afectando al ajuste de distribución de los arcos de esta zona.

Una explicación a este comportamiento podría ser la mezcla de perfiles de tiempo que provoca el semáforo. Un porcentaje de las veces que un vehículo se aproxime al cruce podrá pasar sin detenerse, mientras que el resto, deberá parar y esperar hasta que el semáforo se lo indique. Así, se presentan dos perfiles muy distintos (uno rápido gracias al paso directo, y uno lento producto de la detención) para un mismo (arco,hora) lo que originaría el mal ajuste.

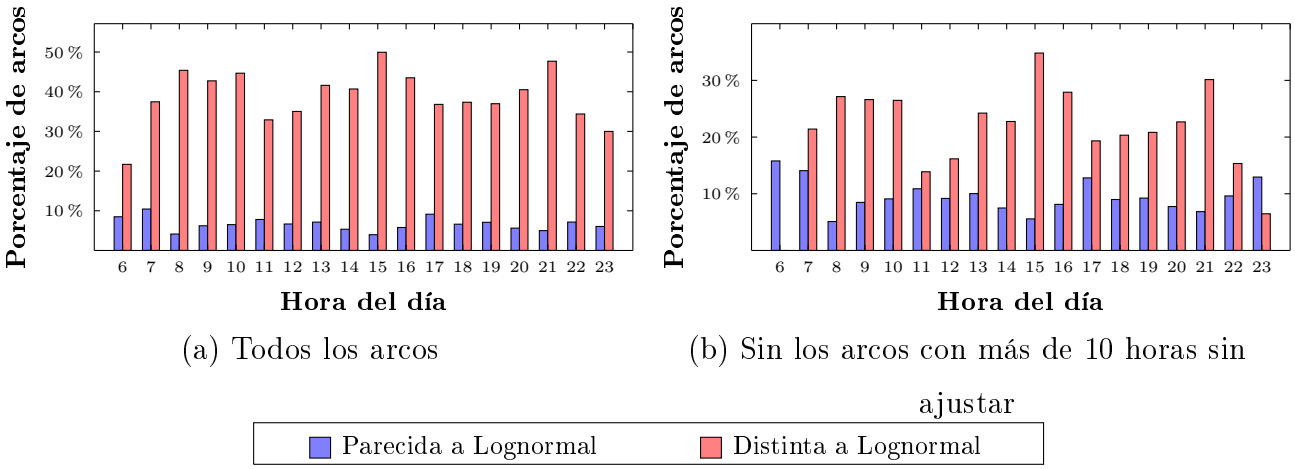
Otro factor externo que también afectó las distribuciones son los resaltos. Estos están marcados con una circunferencia azul en la Figura 6.2. En el cuadro superior, en la esquina derecha, el efecto de uno de estos resaltos es más claro.

Figura 6.2: Ejemplo de arcos conflictivos ($\geq$ 15 horas sin ajustar) se marcan los arcos con problemas, los paraderos, resaltos y semáforos

En la Figura 6.4, a diferencia de los mapas anteriores, se muestran los arcos que en al menos 15 horas tienen un ajuste Lognormal. Se puede apreciar que el efecto de los semáforos nuevamente está presente, aunque esta vez es por su ausencia. En todas las esquinas, aparecen vacíos, lo que es consistente con las afirmaciones que hicieron anteriormente. En este mapa es posible también observar que los segmentos extensos tienen mejores ajustes, probablemente porque las diferencias se compensan en caminos más extensos.

Teniendo estos dos contrastes, se concluye que los paraderos aparecen tanto en los arcos con buenos ajustes como en que no lo tienen, por lo cual es difícil a simple vista clasificarlos en un grupo u otro. Además, en muchos casos la diferencia entre el efecto de éstos con el de los semáforos no queda claro puesto que comparten ubicación geográfica.

Con el fin de estudiar las diferencias en los ajustes de los arcos que son paraderos con los que no, se realizan varios test estadísticos. Los paraderos son segmentos de la ruta del



(a) Todos los arcos

(b) Sin los arcos con más de 10 horas sin ajustar

Figura 6.3: Número de arcos para cada hora que cumplen con tener una distribución muy parecida a una lognormal (azul), y muy distinta (rojo)

bus que han sido identificados en el conjunto de los arcos, usando una variable binaria para ello. La Tabla 6.2 muestra los resultados para estos arcos y los que no son paradas, en siete pruebas. Los primeros tres registros de la Tabla contienen el porcentaje de arcos totales que en 15, 13 y 10 horas de las 18 que se estudian, fallan el criterio de Kolmogorv-Smirnov. Las siguientes 3 pruebas, miden aquellos arcos que ajustan a una distribución Lognormal, en 15, 13 y 10 horas. La última fila miden todos los pares (arco,hora) que fallan con el criterio. La mayoría de los test muestran diferencias no significativas al 95 % entre un grupo y el otro.

Prueba	Proporción paradero(%)	Proporción resto (%)	Superior (95 %)	Inferior (95 %)
Al menos $15 \geq 95\,\%$	8.70	13.86	-11.46	1.12
Al menos $13 \geq 95\,\%$	16.30	18.20	-10.00	5.20
Al menos $10 \geq 95\,\%$	25.00	28.14	-12.64	6.34
Al menos $15 < 95\,\%$	22.83	18.20	-4.48	13.73
Al menos $13 < 95\,\%$	47.83	36.51	0.44	22.19
Al menos $10 < 95\,\%$	64.13	58.10	-4.47	16.54
Todos los que $\geq 95\,\%$	35.67	40.52	-7.48	-2.23

Tabla 6.2: Análisis de arcos paraderos con respecto al resto de los arcos. Se muestran distintas pruebas, con rojo aquellas donde la diferencia es estadísticamente significativa

En los 3 primeros casos la diferencia no es significativa, aunque el porcentaje de arcos aceptados es menor para los paraderos. En las tres siguientes filas, hay una prueba que es significativa y nuevamente los paraderos presentan un porcentaje favorable. Finalmente, al analizar todos los (arco,hora) de cada segmento, la diferencia es significativa, mostrando que los paraderos ajustan mejor.

El resultado a simple vista es poco intuitivo: los arcos de paraderos muestran una leve ventaja sobre el resto de los arcos. Se proponen dos explicaciones para este resultado:

- Los arcos son muy pequeños para abarcar la zona completa que influye el paradero: Para cada paradero se conoce la coordenada en latitud y longitud de su ubicación, la

Figura 6.4: Ejemplo de arcos que al menos tienen 15 horas donde se acepta el ajuste Lognormal. Se marcan los paraderos y los semáforos

cual se proyecta en el arco más cercano del recorrido. El arco, en ocasiones puede no ser lo suficientemente extenso para el paradero, por lo que el efecto que este tiene en las distribuciones de los tiempos de viaje contempla el resto de los arcos.

- Los paraderos generalmente están asociados a esquinas: Se identificó que los semáforos tienen un gran impacto en las distribuciones de los tiempos de viaje, y como éstos comparten lugar con los paraderos, el efecto de los paraderos no es claro.

Con estos resultados se puede continuar utilizando los arcos que son identificados como paraderos para encontrar la distribución de un camino que usa alguno de aquellos arcos. Naturalmente, el incluir o no estos arcos es una interrogante, y gracias a este análisis, es posible establecer que no es un factor clave en las distribuciones.

6.1.2. Caracterización por hora del día

En la Figura 6.3 se despliegan dos gráficos. (a) contiene todos los arcos, incluyendo los que no ajustan bien, mientras que (b) se han quitado los arcos en que no hay ajustes en más de 10 horas. El objetivo es identificar los horarios en los cuales las distribuciones pueden verse afectadas favorable o desfavorablemente.

Para el primer caso, (a), el porcentaje de casos malos, es decir, el total de pares (arco, hora) donde la hipótesis de diferencia con una Lognormal con más de 95 %, bordea el 40 %. En este caso, la hora 6 es la que tiene el menor porcentaje llegando a un 20 %, por lo que presenta menos problemas. Por otra parte, en la hora 15 se alcanza el 50 % del total. Le

siguen la 21 y la 8. Los horarios con mejor comportamiento son la 11 y la 17, quienes se acercan al 35 %. Los buenos casos, aquellos menores a 20 %, se mantienen entre el 5 y 10 % del total, mostrando su mejores horarios a las 6 y 17 horas.

El segundo gráfico, (b), tiene las mismas curvas, una vez que se han excluido los arcos que no ajustan bien. Muestra el mismo patrón que (a), solo que es más notorio el comportamiento que antes se describió. Sin embargo, aquí el perfil de las 15 hrs. no ha mejorado sustancialmente, por lo que probablemente la gran parte de los arcos no puede ajustar esa hora.

Recapitulando, las horas que tienen un efecto combinado entre hora de congestión y hora expedita, como las 21, 8 y 15, tienen el peor comportamiento, mientras que las que tienen mayor homogeneidad en sus tiempos, ajustan mejor. Ejemplo de ello son las 11, 17 y 23 horas. De esta forma, se observa que ciertos bloques horarios están incluyendo patrones de distribución diferentes, lo cual está afectando a las distribuciones.

A continuación, se muestra un caso que permite ilustrar este efecto combinado de distribuciones. En la Figura 6.5 se estudia el factor promedio de ajuste para distintos momentos a lo largo de las 8 hrs. El primer segmento es para los tiempos entre 8:00 y 8:10, el segundo para las 8:00 y 8:20, así hasta llegar a 8:00 y 8:59. Se observa que los primeros 10 minutos son bastante homogéneos, destacando un factor muy bueno.

El gráfico sugiere que se obtendrían mejores resultados si se particionaran las 8 hrs, en dos: desde las 8:00 y 8:30 y las 8:31 y 8:59. Es muy probable que durante la primera mitad los tiempos sean menores que los de la segunda mitad, por lo que es mejor tratar ese bloque segmentado en dos.

En un comienzo se plateó que los segmentos horarios eran arbitrarios y que esto podía afectar las distribuciones, lo cual finalmente, se ve reflejado en los resultados. Con una mayor cantidad de datos por hora, se propone encontrar mejores particiones para el tiempo.

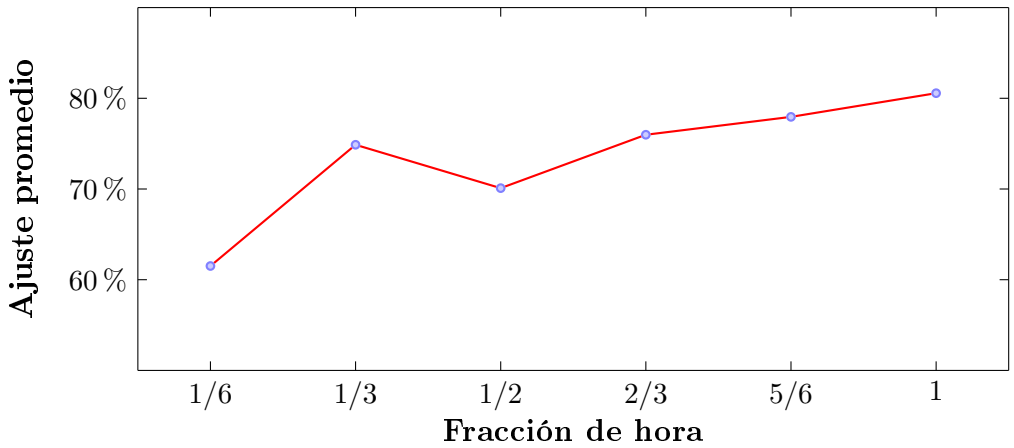


Figura 6.5: Factor de ajuste promedio para distintas particiones de las 8 hrs.

6.1.3. Caracterización por tipo de calle

Se estudia también la incidencia que podría tener el tipo de arco en el comportamiento de la distribución. Este tipo puede ser: avenida, calle, carretera o rotonda. Los mapas anteriores (Figura 6.2, Figura 6.4) muestran diferentes tipos de vías y es posible observar diferencias entre, por ejemplo, avenidas y calles. Esto conduce a pensar que se podría tratar de un factor que afectaría las distribuciones.

Se toma el total de pares arcos por hora que tienen una distribución aceptada como Lognormal, junto con el porcentaje que son muy buenos (al menos 16 horas de ajuste) y además los muy malos (en 16 horas no ajustan), por cada tipo. Esta información se despliega en la Tabla 6.3.

Tipo	Todos los arcos, hora			Total arcos	Al menos 16 horas $\leq 95\,\%$		Al menos 16 horas $> 95\,\%$	
	$\leq 95\,\%$	total	porcentaje(%)		total	porcentaje(%)	total	porcentaje(%)
Carretera	11	39	28.21	3	0	0	0	0
Avenida	7298	11540	63.24	719	84	11.68	48	6.68
Calle	1881	4033	46.64	257	18	7.00	46	17.90
Rotonda	709	912	77.74	57	4	7.02	0	0

Tabla 6.3: Análisis por tipo de arco en el grado de ajuste

Los arcos tipo carretera son muy pocos del total, solo son 3 de 1036. Los arcos que tienen influencia en el total son las calles y avenidas. Las rotondas son un tipo especial de arcos que a veces pueden parecer avenidas y en otros casos calles. Su peso tampoco es significativo, por lo cual las avenidas y las calles van a concentrar la atención.

Las calles tienen bastante menos ajustes que las avenidas. Cuando se toma el total, las avenidas tienen casi 17 puntos porcentuales más que las calles. En efecto, de las calles solo se pueden aceptar un 47 % del total, mientras que de las avenidas un 63 % está bien ajustado. Asimismo, el porcentaje de arcos Lognormales en 16 horas de las 18 que se estudian, es mucho mayor para las avenidas, alcanzando aproximadamente el 12 %. Por otro lado, este porcentaje es de un 7 % para las calles.

Finalmente, aquellos arcos con falla en el criterio en al menos 16 horas de las 18, tienen un impacto mayor en las calles que en las avenidas. Este porcentaje en las cales es de un 18 % aprox. y en las avenidas más de un 10 % menos, lo que permite establecer que las avenidas tienen más arcos que aceptan el criterio para sus distribuciones que las calles.

Este resultado es consistente con lo que se había visto, pues las avenidas tienden a ser más extensas en cuanto a la longitud de los arcos que las conforman, y además, tienen menos factores externos que podrían influir en las distribuciones, tales como semáforos, pasos peatonales, etc.

6.2. Análisis de correlación

Paso siguiente, se quiere estimar cómo se comportaría una secuencia de arcos, conociendo su distribución a partir de los arcos individuales que la conforman. Una suma de variables aleatorias Lognormales, según se trató anteriormente, tiene una distribución que se parece a una del mismo tipo. Sin embargo, los distintos arcos no son independientes y es necesario incluir la correlación existente entre ellos.

Se analizan distintos pares de segmentos y se observa que la correlación promedio para esos arcos separados por una cantidad de arcos dada. En la Tabla 6.4 se muestra la correlación entre arcos separados por 0, 1, 2 y más arcos.

Arcos de distancia	Promedio (%)	Desviación std. (%)	Inferior (95 %)	Superior (95 %)
0	9.26	1.57	6.12	12.41
1	8.51	1.26	5.99	11.04
2	7.91	1.36	5.19	10.64
3	8.78	1.15	6.47	11.09
4	9.28	1.82	5.64	12.92
5	8.98	1.40	6.18	11.78
10	8.43	1.12	6.19	10.68
20	7.50	1.00	5.51	9.49

Tabla 6.4: Correlación promedio para arcos separados por una cantidad de arcos dada

El parámetro es significativo, incluso para arcos muy separados, lo que valida la idea de que es un factor que se debe considerar hasta al menos 20 arcos de distancia. A esta distancia, recién se empieza a ver una baja en la correlación, aunque sigue siendo estadísticamente no diferente a las anteriores. Es más, los 8 valores estimados generan un intervalo de confianza al 95 % que no permite asegurar que uno de ellos sea distinto a cualquier otro.

Cada par de arcos atribuye la correlación que se les calcula, a una serie de filas de datos provenientes de distintos buses. Un bus genera para estos arcos un par de datos que se usan para estimar el factor de correlación, el cual es significativo, según los resultados encontrados.

Si a una hora del día determinada hay congestión es muy probable que a 5, 10 ó 20 arcos más allá la siga habiendo. A las 19 hrs. en general todas las calles o avenidas de la ciudad van a mostrar atochamiento, mientras que a las 6 hrs. en ningún caso lo habrá. Este comportamiento rápido o lento que pueda tener un arco dado, debería mantenerse para los arcos siguientes, explicando que este efecto sea persistente.

Finalmente, se puede fundamentar que la correlación es un factor que debe tomarse en cuenta, el cual, si se incluye, tendrá un impacto en la distribución del tiempo de viaje de una secuencia de arcos. Además, la varianza de este tiempo será mayor puesto que los arcos tienen dependencias positivas.

Para el calculo de esta variable aleatoria resultante, al menos se deben tomar en cuenta los efectos que tienen los arcos cercanos entre sí, en una distancia no menor a 500 m., que es la distancia promedio que cubren 20 arcos.

6.3. Comparación distribución real y estimada

Cuando ya se tienen todos los parámetros estimados, y ya se ha validado la distribución de los tiempos con que se viaja por un determinado arco, se procede a estudiar cómo se comporta la suma de estas variables. Como objeto de referencia se toman los tiempos reales de viaje que extraen de los puntos GPS. Para cada bus se registra el momento en que pasó por el origen del dato, lo que permite conocer con exactitud el tiempo que le tomó a este bus atravesar un segmento dado por varios arcos. Este tiempo aleatorio se compara con el que se estima usando los distintos métodos que aquí se presentan.

El método de Mehta tiene el inconveniente de que es difícil de implementar para un gran número de arcos, por lo que inicialmente se presentan resultados con pocos arcos. Además, según los análisis de los distintos arcos, se determina que las avenidas son las que tienen el mejor comportamiento, siendo estas rutas las que se comparan con los datos reales, inicialmente.

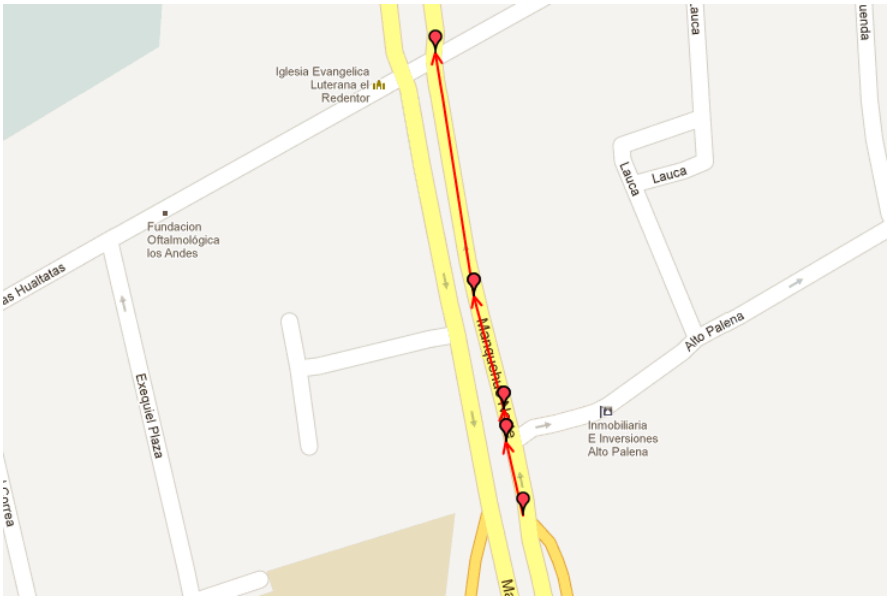
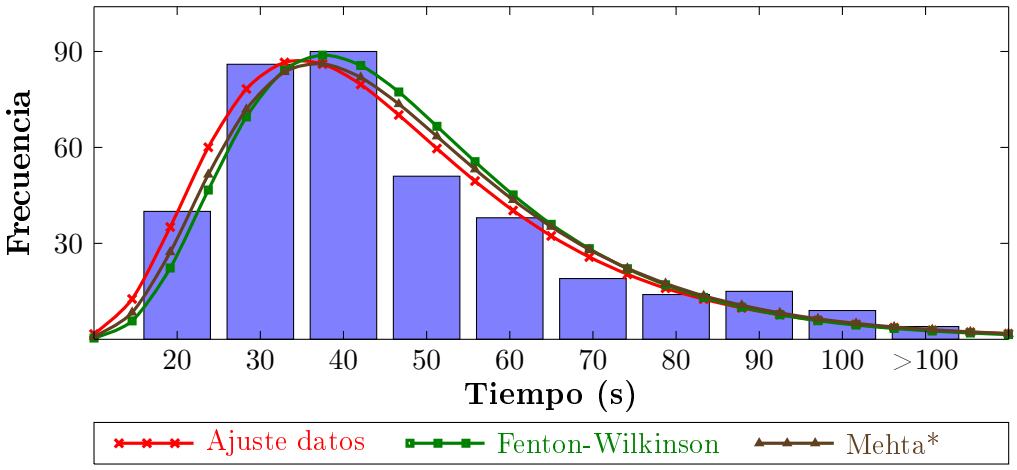


Figura 6.6: Caso 1 de análisis. Segmento de 4 arcos en avenida Manquehue

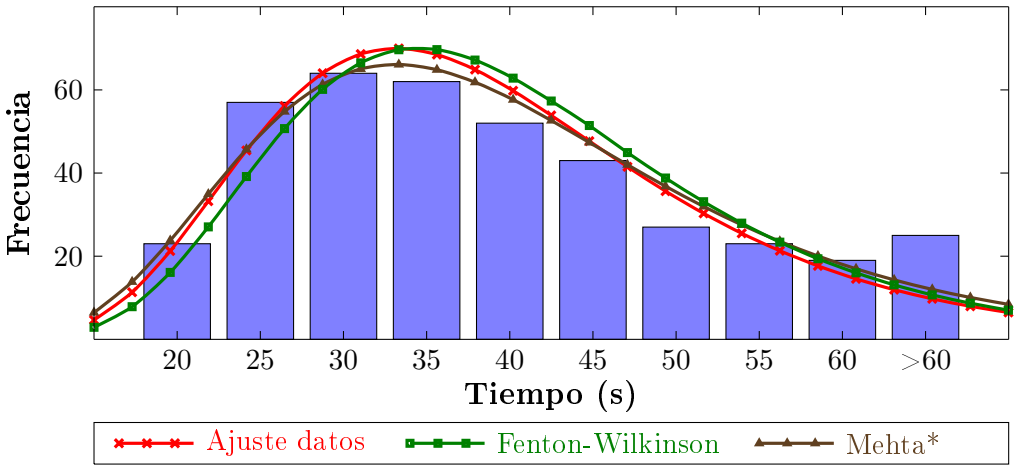
En la Figura 6.6 se muestra un segmento de 4 arcos en una avenida importante de Santiago, el cual es el primer ejemplo que se tomará. Este segmento comprende 190 metros y se analiza en dos horas distintas del día. En la Figura 6.7 se grafican los datos reales y las distintas curvas de ajuste.

Para estimar los parámetros usando el método de Mehta es necesario fijar dos valores s_1 y s_2, los cuales otorgan cierta flexibilidad a las estimaciones. Los autores recomiendan $s_1 = 0{,}0001$ y $s_2 = 0{,}0005$, y para los cálculos que aquí se llevan a cabo se utilizan los mismos números.

Las 3 curvas muestran un comportamiento parecido para ambos gráficos, aunque los ajustes estimados son mejores para (b), el histograma de las 10 hrs. Con rojo se representa la curva que generan los datos, es decir, se estiman tanto μ como σ provenientes de los puntos GPS. Para estos valores, el comportamiento de los datos queda muy bien descrito, lo cual



(a) Histograma tiempo de viaje para las 18 hrs. Ajuste de 44.34 % para 366 datos



(b) Histograma tiempo de viaje para las 10. Ajuste 53.92 % para 395

Figura 6.7: Histograma tiempo de viaje para dos distintas horas del día para el segmento de arcos del Caso 1. Se señala el ajuste determinado por los datos, el método de FW y el método de Mehta

queda reflejado en el factor de ajuste, un 44.34 % para las 18 hrs. y un 53.92 % para las 10 hrs.

Lamentablemente, la curva que estima FW en los dos gráficos, está desplazada hacia la derecha, lo que refleja que en algún grado, la varianza estimada por este método es menor a la que se estiman los otros métodos, y, consecuentemente, la media está sobrestimada.

En la Tabla 6.5 se resume la información de los gráficos para este segmento. Se muestra que, efectivamente, tanto μ como σ están sistemáticamente sobre y subestimados, respectivamente. Este error impacta sobre el factor de ajuste, provocando que este método no pueda estimar una curva que sea aceptada por el criterio planteado.

Por otro lado, el método de Mehta, estima parámetros que ajustan mejor y que no tienen un patrón de error como FW.

	Ajuste	μ	σ	factor (%)
(a)	Ajuste de datos	3.7465	0.4429	44.34
	Estimación por FW	3.7932	0.4062	98.80
	Estimación Mehta	3.7844	0.4276	96.38
(b)	Ajuste de datos	3.6192	0.3419	53.92
	Estimación por FW	3.6496	0.3302	98.78
	Estimación Mehta	3.6365	0.3681	75.20

Tabla 6.5: Parámetros estimados para segmento del Caso 1

	Datos	Mehta	Fenton-Wilkinson
Promedio factor(%)	71.71	78.11	89.59
Porcentaje de Lognormales(%)	80.00	63.33	40.00
Condicional (%)	-	66.67	37.50
Promedio $\sqrt{(\sigma_{datos} - \sigma)^2}$ (%)	-	16.44	10.08
Porcentaje diferencia positiva σ (%)	-	36.67	100
Promedio $\sqrt{(\mu_{datos} - \mu)^2}$ (%)	-	10.07	12.30
Porcentaje diferencia positiva μ (%)	-	63.34	0

Tabla 6.6: Resumen de ajustes por distintos métodos para segmentos de cuatro arcos

Experimentos de este tipo se repitieron varias veces y luego se construyó una tabla resumen con información agregada de los resultados. En la Tabla 6.6 se resume los resultados obtenidos por los métodos de estimación de μ y σ para diferentes segmentos con 4 arcos consecutivos. El total de veces que se estimaron tiempos de viaje en caminos para construir esta tabla es de 30, es decir, se tomaron 30 caminos en diferentes lugares del mapa en distintos horarios, lo que origina esta tabla.

En promedio, los datos ajustan mejor que los otros métodos, aunque no quedan significativamente alejados de lo que estima Mehta. FW, en cambio, bordea el 90 % por que muchas de sus estimaciones quedan fuera de lo que realmente debería estimar.

Un 80 % del total, puede considerarse como una variable aleatoria Lognormal, porcentaje que llega casi al 64 % usando Mehta y un 40 % usando FW. Esto quiere decir que usando los datos reales, este porcentaje refleja cuando el método para estimar permite explicar la curva que se construye. Se deduce que del 80 % que tiene una distribución Lognormal, Mehta permite explicar más de un 60 % y que FW solo un 40 %.

Si se toma el ajuste condicional de los dos métodos, es decir, el porcentaje de ajuste de Mehta y FW, dado que los datos ya muestran la distribución estudiada, se observa que el primero cubre 2/3, mientras que FW, solo la mitad de este porcentaje.

Dos explicaciones que son consistentes entre sí, tienen origen en los valores de μ y σ estimados. Para Mehta el valor de σ a veces es mayor que el valor estimado por los datos, y otras veces es menor. Un 36.67 % es menor, y el porcentaje restante, es mayor. En cambio, FW muestra sistemáticamente un valor menor, siendo el 100 % de veces más pequeño que el real, lo cual es coherente con lo visto en el Caso 1.

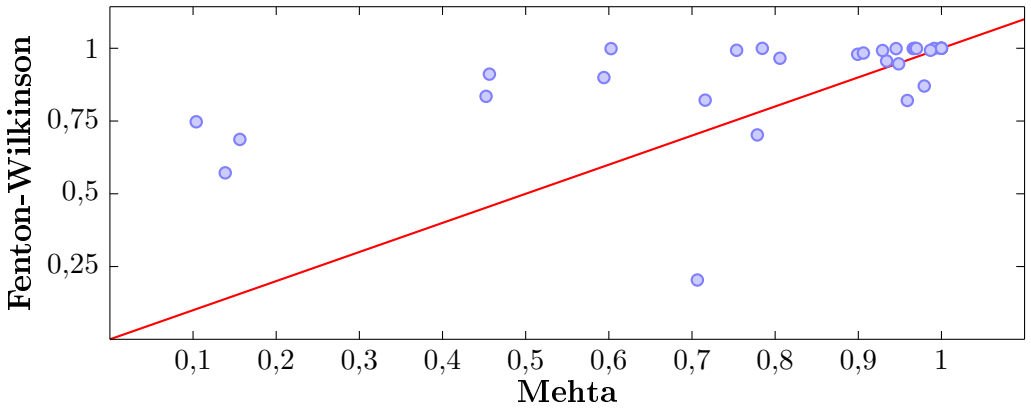


Figura 6.8: Gráfico de comparación de ajustes por métodos distintos, FW y Mehta

El efecto inmediato es que el μ debe compensar el mayor o menor ajuste de σ. Es por esta razón que este parámetro está en el 100 % de los casos sobre estimado para FW, mientras que, es solo un 63.34 % para Mehta.

En general, estos distintos enfoques encuentran correctamente el valor esperado de viaje para una secuencia de arcos. Recordando que la esperanza viene dada por $\exp(\mu + \sigma^2/2)$, un valor más bajo de σ debe ser compensado con un valor mayor de μ.

Las dificultades que se presentan en FW, y que no muestra Mehta, permiten entender por qué un método sería preferible a usar en vez del otro. La Figura 6.8 revela esta afirmación. Cada punto del gráfico es un par ajuste FW y ajuste Mehta para los 30 experimentos que se habían comentado. El ajuste está determinado por el criterio Kolmogorov - Smirnov que se ha usado para aceptar o rechazar los resultados.

La línea roja es tal que $ajuste_{Mehta} = ajuste_{FW}$, entonces, cada punto que esté sobre esta línea implica que los dos métodos explican de la misma forma los datos observados, pues presentan el mismo factor. Puntos sobre la línea muestran que el método de Mehta es mejor que el de FW para ese experimento, y bajo la linea, que es peor. Se puede apreciar que más puntos están sobre la línea respaldando el enfoque de Mehta.

Sin embargo, todos los resultados mostrados hasta ahora son de experimentos de no más de 4 arcos consecutivos, lo cual no exhibe contratiempos para ningún enfoque. Mehta, como se trató anteriormente, no puede calcular parámetros para más de 6 arcos en un tiempo razonable. 25 o más arcos hacen inviable el método incluso en muy buenos pc, ya que el número de términos que se deben calcular crece exponencialmente. Por lo tanto, se planteó una manera de enfrentar estos casos, usando una modificación de Mehta, Mehta*.

Un ejemplo de este tipo, es el Caso 2, un segmento de arcos 22 en dos avenidas distintas. La Figura 6.9, grafica el recorrido estudiado.

Están comprendidos aproximadamente 1300 metros en esta secuencia, para la cual se cuenta con todos los parámetros μ_{i} y σ_{i} para cada arco $\mathrm{i} \in Arcos$, y cada correlación $\rho_{\mathrm{i}j}$ entre todos los pares de este conjuntos, $(\mathrm{i}, j) \in Arcos^2$.

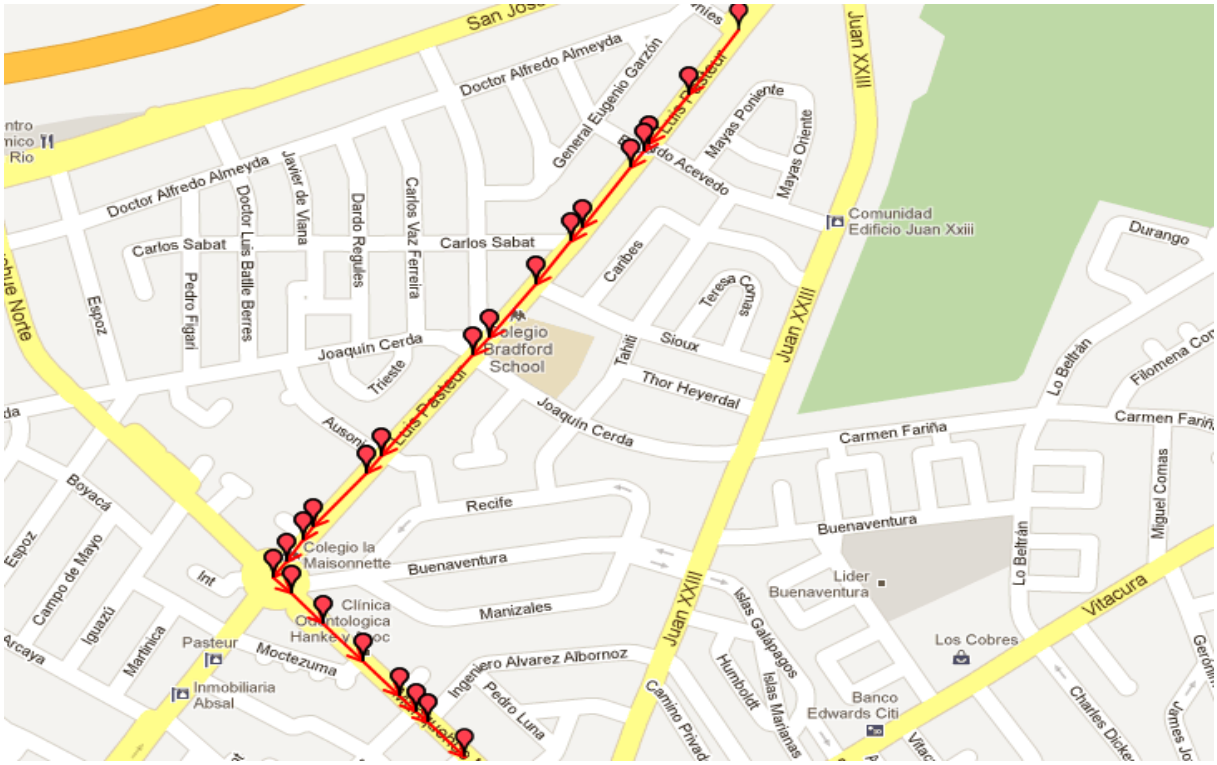


Figura 6.9: Caso 2 de análisis. Segmento de 22 arcos que describen el viaje por dos avenidas

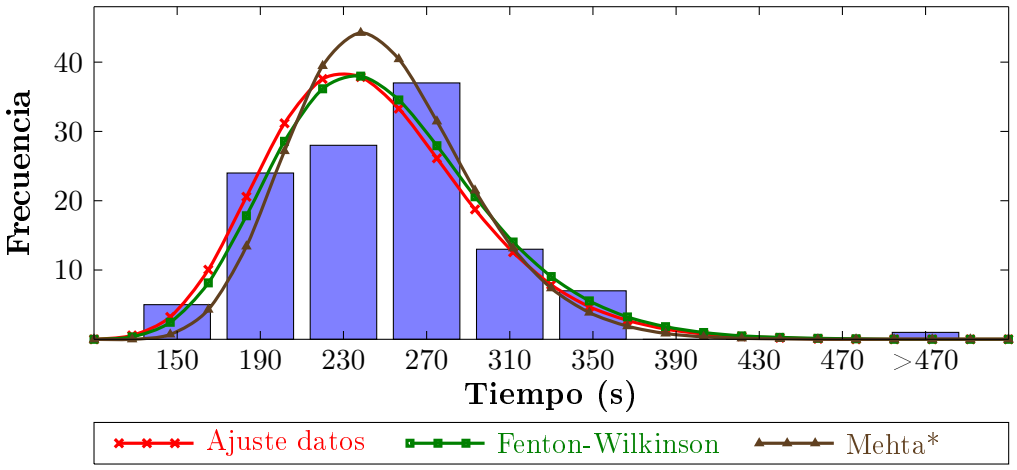
Se analizan dos horarios, las 20 hrs. y las 13 hrs., las cuales fueron escogidas de manera arbitraria. A simple vista, se aprecia lo bien que ajustan los datos, representados en el histograma de la Figura 6.10, con una distribución Lognormal. La curva roja, corresponde a la curva proveniente de estos datos, para la cual el factor de ajuste es de casi un 14 % en las 20 hrs., y un aproximado 20 % para las 13 hrs. El buen ajuste para este segmento respalda la idea que se comentó anteriormente: segmentos más extensos tienen mejores ajustes puesto que las diferencias producidas por factores externos se compensan en tramos más largos.

Usando FW con correlaciones y Mehta modificado para segmentos de 4 arcos, se estiman las curvas que acompañan al histograma. A diferencia de lo ocurrido en el Caso 1, todos los métodos alcanzan a clasificar, y de manera razonable, explican los datos observados. No obstante, la menor varianza no es para el método de FW, sino para Mehta*, lo cual se advierte en el pick al que llega la curva de Mehta* en ambos gráficos.

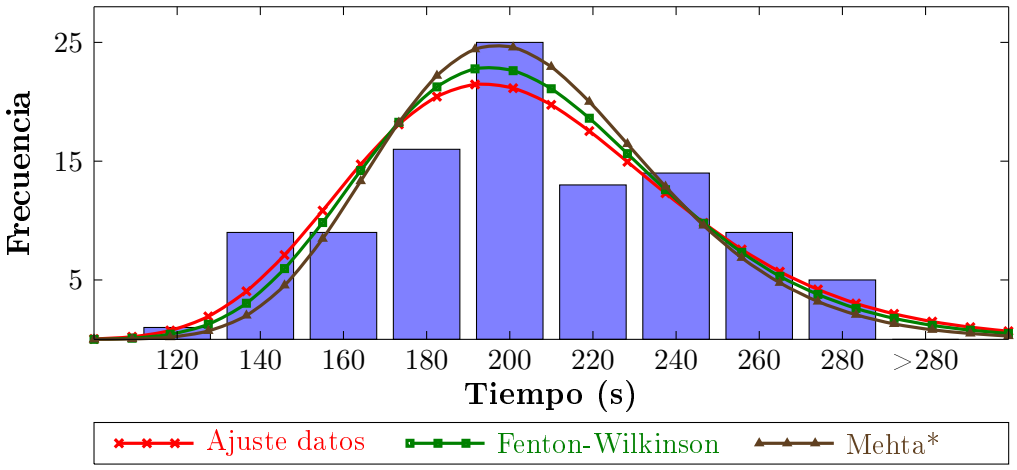
	Ajuste	μ	σ	factor (%)
(a)	Ajuste de datos	5.4793	0.2030	13.69
	Estimación por FW	5.4989	0.2010	46.79
	Estimación Mehta*	5.5046	0.1712	93.42
(b)	Ajuste de datos	5.3043	0.1917	19.52
	Estimación por FW	5.3068	0.1784	42.49
	Estimación Mehta*	5.3095	0.1635	68.03

Tabla 6.7: Parámetros estimados para segmento del Caso 2

En efecto, aparentemente la curva de FW es mejor en este caso, acercándose más a lo



(a) Histograma tiempo de viaje para las 20 hrs. Ajuste de 13.69 % para 115 datos



(b) Histograma tiempo de viaje para las 13 hrs. Ajuste de 19.52 % para 101 datos

Figura 6.10: Histograma tiempo de viaje para dos distintas horas del día para el segmento de arcos del Caso 2. Se señala el ajuste determinado por los datos, el método de FW y el método de Mehta*

esperado que Mehta*. La Tabla 6.7 tiene esta información y reafirma lo que muestran los gráficos: FW ajusta mejor aunque muestra la misma tendencia que antes, sobrestima μ y da un valor más pequeño a σ.

A pesar de que los dos enfoques tienen un error importante, los ajustes son aceptables, explicando bastante bien la curva de datos para este caso.

Se analiza, también, qué ocurre cuando el número de arcos es mayor. Tomando una secuencia de 37 arcos, se replican los cálculos que se hicieron para los casos anteriores. Los arcos analizados se encuentran graficados en la Figura 6.11, correspondiente al Caso 3.

Nuevamente se trata de dos arterias importantes de Santiago, en las cuales se presentan 4 semáforos. Se toman dos horas distintas del día, las 9 hrs. y las 20 hrs, cada uno con al menos 90 datos. Corresponden a dos horas de bastante congestión, lo cual no impide que el

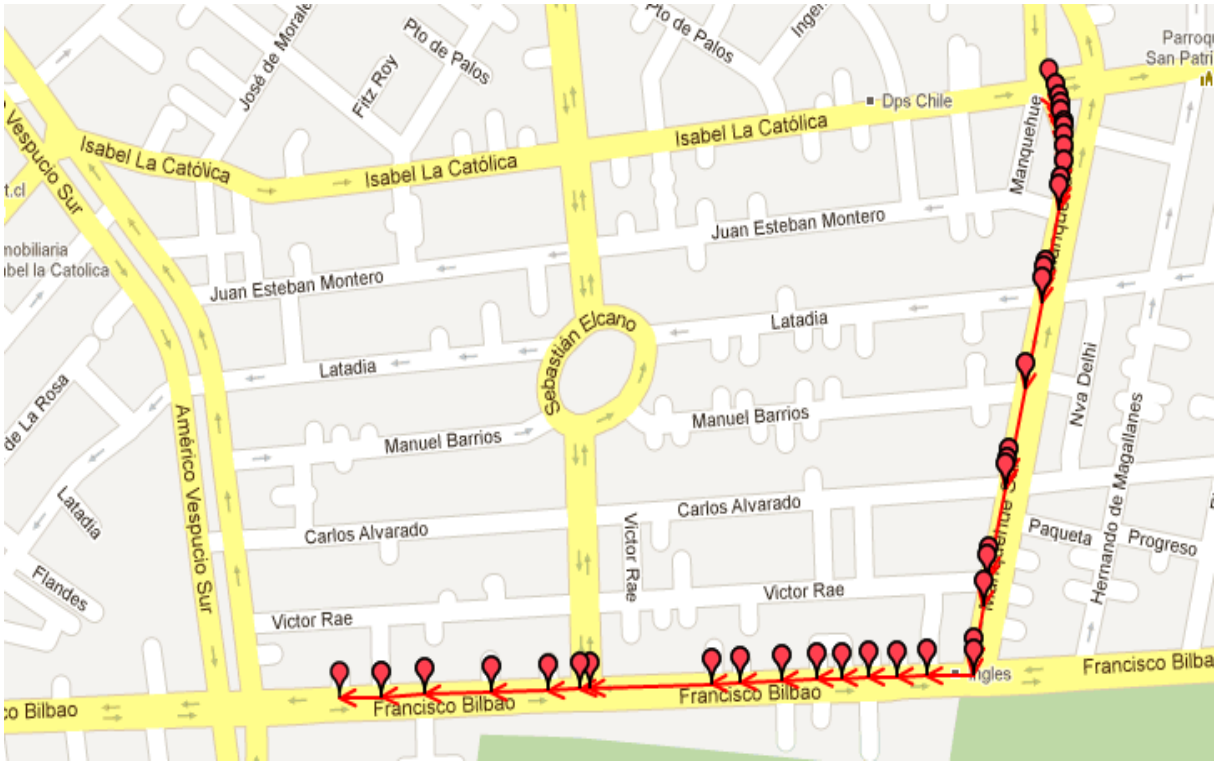


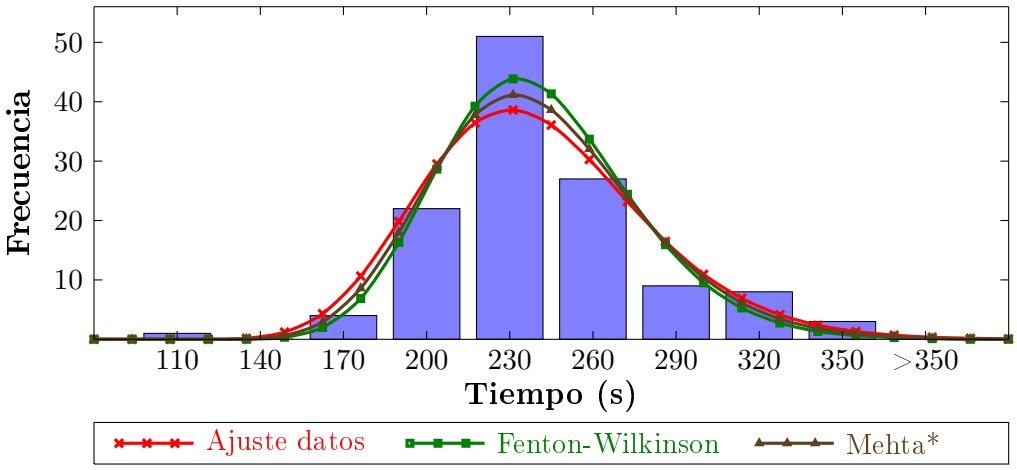
Figura 6.11: Caso 3 de análisis. Segmento de 37 arcos que describen el viaje por dos arterias importantes de Santiago

ajuste sea aceptable y que ambas curvas se permitan decir que tienen un comportamiento Lognormal. El tiempo promedio de viaje es mayor para las 20 hrs., sin embargo, la curva para las 9 hrs. es mejor catalogada por el criterio. Estos datos están graficados en la Figura 6.12.

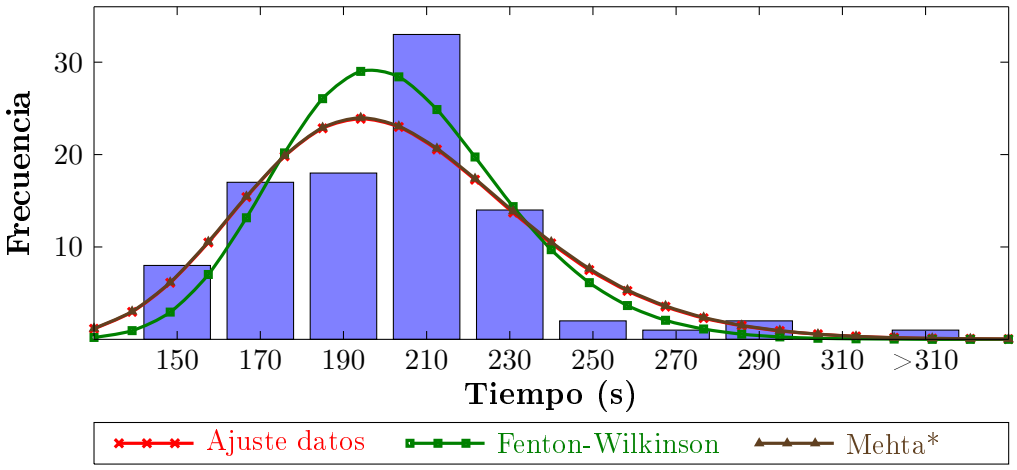
	Ajuste	μ	σ	factor (%)
(a)	Ajuste de datos	5.4669	0.1667	74.80
	Estimación por FW	5.4714	0.1450	60.13
	Estimación Mehta*	5.4701	0.1551	58.63
(b)	Ajuste de datos	5.2961	0.1636	51.13
	Estimación por FW	5.3011	0.1329	60.63
	Estimación Mehta*	5.2967	0.1644	54.25

Tabla 6.8: Parámetros estimados para segmento del Caso 3

Un fenómeno extraño que ocurre para el histograma de (a) es que el ajuste de datos es peor que el criterio para los dos métodos, los cuales están señalados en la Tabla 6.8. Casi un 75 % de los datos, es superado por el aproximado 60 % de FW y el un poco menor 58.63 % de Mehta*. ¿Cómo se explica que los parámetros provenientes de los datos sean los peor evaluados? Se debe recordar que estos parámetros no son los máximos verosímil, es decir, aquellos que maximizan la probabilidad de lo que se está observando, sino, más bien, son los que estiman insesgadamente los valores de la distribución. Entonces, es posible que los métodos encuentren mejores valores para el criterio establecido, aunque estos no deberían ser muy distintos a los insesgados.



(a) Histograma tiempo de viaje para las 20 hrs. Ajuste de 74.8 % para 125 datos



(b) Histograma tiempo de viaje para las 9 hrs. Ajuste de 51.13 % para 96 datos

Figura 6.12: Histograma tiempo de viaje para dos distintas horas del día para el segmento de arcos del Caso 3. Se señala el ajuste determinado por los datos, el método de FW y el método de Mehta

Todos los resultados indican un buen ajuste, a pesar de que comprende 37 arcos. El mejor método es el de Mehta*, superando a FW en los dos horarios, aunque la diferencia entre los parámetros es baja. Esta diferencia es poco perceptible al momento de mirar μ o σ, pero sí es notoria en el factor de ajuste.

Cabe destacar que el comportamiento del método de FW persiste para este caso. Enseguida, μ estimado por este enfoque es el mayor de todos, tanto para (a) como para (b), mientras que σ, es el menor en los mismos escenarios.

Un hecho a tomar en cuenta, es la gran precisión con que Mehta* logra encontrar la curva para esta variable. Es tanto así, que la diferencia con la curva de datos no es visible en el histograma (b) para este caso.

6.4. Horarios similares en distribución

Una de las preguntas que queda pendiente es: ¿Qué horarios se pueden agrupar en uno solo? Es decir, ¿hay manera de que el día se pueda dividir de otra forma? Por ejemplo, ¿agrupar las 6 con las 7? ¿o las 18 y 19?

Para comenzar a responder esas preguntas se hace el siguiente análisis: por arco, se toman todos los datos de velocidades entre las 6:00 y 23:59 hrs., y posteriormente, se calcula la velocidad promedio de cada una de esas horas. Para cada arco, entonces, se tiene un valor promedio de velocidad en cada una de las 18 horas. Con estos promedios, se construye un orden de mejor a peor hora para desplazarse por tal arco, lo cual que permite construir un ranking de velocidad para cada segmento. Tomando las cuatro peores horas del día y las cuatro mejores, se conforman las barras del histograma de la Figura 6.13. De esta forma es posible representar las horas del día con mayor congestión, y las con menor. A simple vista, las barras azules están presente principalmente en la tarde y parte de la mañana, los horarios con mayor afluencia de autos por las calles. En cambio, las barras rojas muestran los momentos del día que no tienen congestión, siendo el más rápido el de la mañana.

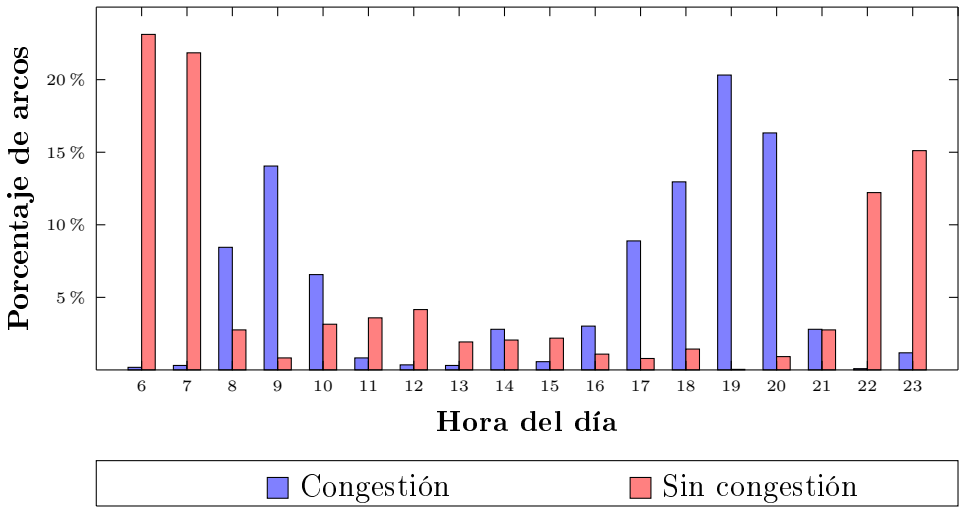


Figura 6.13: Histograma de horarios de congestión vehicular y sin congestión

Tanto las 6 como las 7 de la mañana son muy buenos horarios para moverse por Santiago. Un poco más atrás le siguen las 22 y 23 hrs. Por otra parte, se encuentran las 19 y 20 hrs., horarios bajo los cuales es muy lento desplazarse, siendo el primer intervalo el peor para la mayoría de los arcos. Las 9 y 18 hrs. les siguen como momentos del día con alta congestión.

A modo general, se tiene un bloque de horario congestionado en la mañana 8-9-10 hrs., y uno más extenso en la tarde noche, entre las 17 y 20 hrs. Los horarios muy temprano y muy tarde son sin duda los mejores. Finalmente, entre las 11 y 16 hrs., pareciese que hubiera un horario mixto, sin tendencias claras.

Este perfil, permite de antemano, seleccionar candidatos a pertenecer a un mismo grupo, donde uno de los más marcados estaría entre las 6 y 7 hrs. La Figura 6.14 complementa esta idea. Aquí se muestra una matriz de distancias de distribución empírica por hora del día,

distancia dada por el test de Kolmogorov-Smirnov para dos muestras, el cual se explica en el Capítulo 5.

	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
6	-	57.75%	76.99%	81.92%	76.94%	75.27%	75.74%	78.88%	81.58%	79.63%	82.24%	85.46%	87.76%	90.28%	89.00%	80.83%	71.29%	64.13%
7	57.75%	-	85.21%	90.77%	88.33%	88.93%	89.72%	90.12%	93.10%	91.53%	95.59%	94.61%	94.97%	96.66%	94.90%	93.45%	90.36%	76.75%
8	76.99%	85.21%	-	64.58%	70.31%	79.25%	75.94%	77.96%	80.79%	77.39%	81.20%	85.09%	82.70%	90.43%	87.65%	84.15%	87.64%	81.83%
9	81.92%	90.77%	64.58%	-	66.24%	70.42%	78.86%	76.97%	77.46%	75.41%	75.86%	82.09%	82.31%	88.74%	85.01%	81.04%	85.86%	83.43%
10	76.94%	88.33%	70.31%	66.24%	-	68.45%	73.45%	79.18%	77.00%	73.95%	76.26%	84.30%	82.94%	91.64%	88.78%	85.85%	84.30%	82.34%
11	75.27%	88.93%	79.25%	70.42%	68.45%	-	63.20%	68.51%	69.33%	65.92%	73.29%	81.27%	84.39%	88.09%	85.40%	80.77%	80.92%	80.02%
12	75.74%	89.72%	75.94%	78.86%	73.45%	63.20%	-	66.40%	73.30%	66.68%	71.72%	78.62%	83.11%	87.95%	84.08%	79.64%	81.43%	79.42%
13	78.88%	90.12%	77.96%	76.97%	79.18%	68.51%	66.40%	-	55.14%	60.81%	58.41%	74.52%	77.77%	84.26%	79.75%	71.37%	81.90%	82.36%
14	81.58%	93.10%	80.79%	77.46%	77.00%	69.33%	73.30%	55.14%	-	58.42%	58.83%	73.83%	76.30%	80.76%	80.13%	78.84%	84.25%	82.59%
15	79.63%	91.53%	77.39%	75.41%	73.95%	65.92%	66.68%	60.81%	58.42%	-	55.31%	70.44%	73.68%	82.31%	81.71%	75.59%	84.60%	78.84%
16	82.24%	95.59%	81.20%	75.86%	76.26%	73.29%	71.72%	58.41%	58.83%	55.31%	-	68.59%	74.27%	79.85%	80.78%	75.55%	82.04%	83.01%
17	85.46%	94.61%	85.09%	82.09%	84.30%	81.27%	78.62%	74.52%	73.83%	70.44%	68.59%	-	60.58%	72.83%	77.63%	78.47%	85.20%	86.16%
18	87.76%	94.97%	82.70%	82.31%	82.94%	84.39%	83.11%	77.77%	76.30%	73.68%	74.27%	60.58%	-	62.81%	66.63%	76.08%	89.56%	88.04%
19	90.28%	96.66%	90.43%	88.74%	91.64%	88.09%	87.95%	84.26%	80.76%	82.31%	79.85%	72.83%	62.81%	-	59.97%	91.41%	94.47%	92.20%
20	89.00%	94.90%	87.65%	85.01%	88.78%	85.40%	84.08%	79.75%	80.13%	81.71%	80.78%	77.63%	66.63%	59.97%	-	84.84%	91.58%	92.12%
21	80.83%	93.45%	84.15%	81.04%	85.85%	80.77%	79.64%	71.37%	78.84%	75.59%	75.55%	78.47%	76.08%	91.41%	84.84%	-	72.35%	76.43%
22	71.29%	90.36%	87.64%	85.86%	84.30%	80.92%	81.43%	81.90%	84.25%	84.60%	82.04%	85.20%	89.56%	94.47%	91.58%	72.35%	-	63.99%
23	64.13%	76.75%	81.83%	83.43%	82.34%	80.02%	79.42%	82.36%	82.59%	78.84%	83.01%	86.16%	88.04%	92.20%	92.12%	76.43%	63.99%	-

Figura 6.14: Matriz de distancia promedio de la distribución empírica entre los mismos arcos para horarios cruzados, obtenidos por el criterio de Kolmogorov-Smirnov

Se calcula para todos los arcos el estadístico para cada par de horas de la matriz, estadísticos que se promedian usando los datos de toda la muestra. La diagonal naturalmente, tendrá solo ceros y el resto valores que reflejan la proximidad estadística entre las distribuciones.

Los candidatos antes mencionados, 6 y 7, son los segundos más próximos entre sí, únicamente superados por 13 y 14. Se conforman grupos para los cuales el criterio promedio no debe superar el 70 % y se establecen clusters de horas similares. El siguiente paso es probar qué tan bien resulta agrupar estos horarios y si los resultados de ajuste individual por arcos son mejores que sin agrupar.

En la Tabla 6.9 se muestran resultados que permiten validar la hipótesis o no. Primero se evalúa el porcentaje de los arcos que ajustan para cada horario. Tanto las 6 como las 23 hrs., no tienen el cálculo requerido, puesto que existen pocos arcos que tengan estas horas con suficientes datos. Notar que estos porcentajes son consistentes con los gráficos de la Figura 6.3, la peor hora es la 15, seguida por la 8, mientras que las 22 y las 12 son muy buenas.

	6	7	8	9	11	12	13	14	15	16	18	19	20	22	23
Total individual	-	62.26	54.95	58.92	65.81	65.16	60.65	60.43	52.26	55.06	63.33	64.09	58.49	66.67	-
Total juntos	57.31		33.66		43.98		15.05				30.32			59.68	
Ningún ajuste	4.27		0		0		0				0			1.61	
Un ajuste	89.46		4.80		8.02		0				0.88			88.71	
Dos ajustes	-		72.77		75.00		0				13.64			-	
Tres ajustes	-		-		-		5.74				61.05			-	
Cuatro ajustes	-		-		-		46.55				-			-	

Todos los valores están en porcentajes

Tabla 6.9: Resultados de grupos de horarios, los cuales se han identificado como similares

La siguiente fila corresponde al total cuando se juntan los datos para un mismo grupo, según los grupos que se han conformado. Ningún grupo alcanza a explicar la cantidad de ajustes que tienen sus horas por separado. Es decir, el total de arcos que se explican bien usando la distribución Lognormal, disminuye cuando se agrupan las horas.

Una de las principales razones por la cual se decide juntar estas horas, consistía en que de esta manera se lograra explicar un porcentaje importante de las horas que quedaban fuera

de forma individual. Las siguientes filas, muestran cuántos buenos ajustes se logran si se tiene ningún ajuste de forma individual, un ajuste, dos, así sucesivamente. Se observa que el porcentaje que se gana es muy pequeño, en el mejor de los casos se llega al 5 % de los casos totales. En otras palabras, si ninguna de las horas estudiadas individualmente lograba ajustar, un pequeño porcentaje podrá arreglar esta situación una vez que se agrupan los horarios.

Un grupo que definitivamente no debe juntarse es el de la tarde, 13, 14, 15 y 16 hrs. Incluso cuando dos de las cuatro horas que se estudian ajustan bien, no logra explicar la curva de datos. Cuando se toman todos los casos en que las cuatro horas tienen la distribución deseada, solo un 47 % presenta un ajuste aceptable.

En resumen, no es recomendable agrupar horas en bloques, los ajustes en arcos que no lograban ser explicados no compensan las pérdidas en los arcos que sí lo son individualmente. Implícitamente, estos resultados revelan que los horarios son muy distintos unos de otros, a pesar de los símiles que se presentaba en la Figura 6.14. Posiblemente el análisis que se deba hacer, cuando se incluyan más datos, es al revés, estudiar las horas disgregadas aún más.

Conclusión

Como ya muchos autores lo habían notado, la distribución Lognormal permite explicar bastante bien el tiempo de viaje a través de un arco del grafo de Santiago. Cuando se analiza el total de pares arcos - hora, se logra ajustar bien un 60 %, porcentaje que sube hasta prácticamente un 80 % al momento de quitar aquellos arcos que tienen patrones muy extraños de comportamiento, es decir, que presentan problemas en varias horas del día. Este es un porcentaje alto considerando que el test usado es bastante exigente.

Un grupo muy importante de los arcos estudiados tiene ajustes aceptados en la mayor parte del día, en este caso, en al menos 10 de las 18 horas de estudio, lo que permite afirmar que la distribución escogida modela adecuadamente los tiempos para este gran grupo. Sin embargo, otro segmento de arcos tienen perfiles con un comportamiento que no sigue la distribución Lognormal la mayor parte del día. Tanto estos arcos como los que ajustan bien se caracterizan, lo que permite concluir cuáles son los mejores arcos para explicar con esta distribución.

En primer lugar se identificó que las avenidas son mejores en sus ajustes que las calles. Esta diferencia alcanzaba incluso 20 puntos porcentuales en el total. Es decir, del 100 % de los casos aproximadamente un 40 % de las calles tienen perfiles aceptables, mientras ese porcentaje para las avenidas es de un 60 %. Se puede atribuir esta diferencia a los fenómenos externos que tienen las calles. Dentro de éstos están especialmente los pasos de peatones, resaltos o speed bump, cedas el paso y discos pare, los cuales se encuentran comúnmente en las vías de estas características.

Fenómenos trasversales tanto para calles como avenidas, son los semáforos, los cuales afectan en la mayoría de los casos a las distribuciones. Posiblemente, debido a que un semáforo a veces puede permitir el paso a un vehículo y otras veces no, la distribución queda sugeta a dos perfiles distintos que explicarían los malos ajustes. Además, como en gran parte de los casos los paraderos se relacionan con las esquinas con semáforos, no se encontraron diferencias entre los arcos identificados como paraderos y el resto de la muestra.

Junto con crear un perfil para los arcos, se estudiaron las horas y sus ajustes, lo que permite establecer que las horas con cambios en la congestión vehícular muestran peores resultados que aquellas con un patrón homogéneo. Ejemplo de ello, son las 8 am. y las 21 pm. dos horarios con muy malos ajustes, que durante la primera fracción de la hora tienen un comportamiento y para la segunda, tienen otro. Claramente este tipo de distribución no se puede explicar tan solo con una Lognormal simple. El contraste lo presentan las 6, las 11 y las 23 hrs., todos horarios muy homogéneos que muestran perfiles muy predecibles.

Se analizaron dos métodos para sumar variables aleatorias Lognormales, FW y Mehta. Aunque tienen diferencias, ambos métodos permiten encontrar la distribución para un camino de varios arcos. Los resultados de la suma de los tiempos de viaje, entregan información suficiente para estudiar modelos de ruteo. No obstante, fue necesario incluir correlaciones entre los distintos arcos para que la varianza obtenida reflejara la real. Esta correlación es relevante incluso a más de 500 metros, siendo un factor clave en la suma de estas variables.

Ahora, si bien ambos métodos son aceptables, el método de Mehta se destaca como la mejor alternativa. Un primer análisis deja en evidencia un problema en el método de FW que otros autores ya habían mostrado en sus estudios: la tendencia a estimar una varianza menor a la suma de variables. En los más de 30 casos estudiados para 4 arcos, en el 100 % de los casos FW estima valores de σ menores al que se estima usando los datos, lo cual provoca que μ esté en todos esos casos sobrestimado.

Los ajustes para Mehta, en la mayoría de los casos, son mejores y permiten explicar muy bien caminos en el grafo. Además, cuenta con una variable que le otorga flexibilidad y permite calibrarlo según el caso. En este estudio se utilizaron valores recomendados por los autores, los cuales mostraron ser adecuados.

Se sugiere utilizar Mehta para la suma de tiempos, y para casos en que el cálculo se hace intratable, en este estudio se entrega un método para enfrentar este problema. Se recomienda agrupar los arcos en segmentos que se pueden sumar y luego, restablecer una matriz de correlaciones. Este método mostró precisión en su estimación, junto con manejar caminos formados por varios arcos.

El sistema propuesto mostró buenos resultados, explicando en gran parte los datos que se observaban para los arcos escogidos.

Finalmente, se estudiaron las distribuciones en los distintos horarios, con el fin de fijar bloques en los cuales fuese posible asumir una sola distribución. Los resultados mostraron que la agrupación de horas no ayuda a ajustar en mejor medida los tiempos, y que puede ser recomendable justamente lo contrario, desagregar aún más las horas del día.

Trabajos Futuros

Si bien los tiempos de viaje arrojaron muy buenos ajustes usando la distribución Lognormal, algunos de los horarios estudiados presentaron problemas. En particular, se llegó a la conclusión que los horarios con mezcla de comportamiento vehicular, eran los con mayores dificultades. La principal razón es que una primera parte de la hora estudiada tiene tiempos, ya sea más lentos o más rápidos, que la segunda mitad.

En la Figura 6.15 se despliegan algunos ejemplos que ayudan a entender esta idea. El histograma en (c) es un ejemplo a las 8 hrs., el cual es un horario crítico del día. Aquí claramente se presentan dos Lognormales, siendo la de más a la izquierda probablemente la que tenga un mayor peso dentro del mixture. Esto significa que los tiempos más cortos tienen más importancia a esta hora.

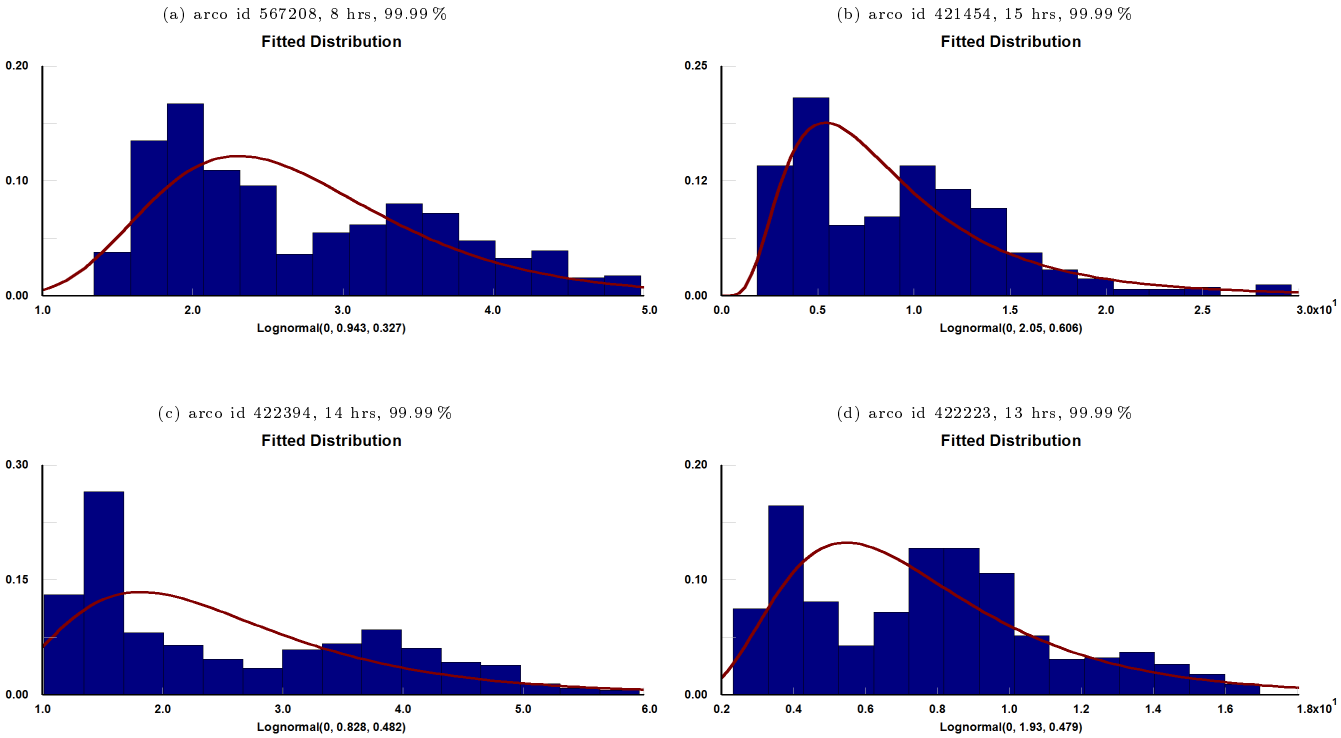


Figura 6.15: Ejemplos de malos ajustes donde se propone mixture de lognormales

Un desarrollo posterior sugiere estimar distribuciones de un mixture de al menos dos Lognormales, lo que ayudará significativamente a incorporar estos tiempos a la clasificación de aceptados. Además, gracias a que se mantiene la estructura de Lognormales, la suma de estos mixture resultará en un mixture de suma de Lognormales individuales, sumas que pueden calcularse fácilmente usando los métodos que se han presentado.

Como tema independiente, que también es interesante de abordar, está el manejo de correlaciones para caminos de muchos arcos usando Mehta. Se presentó un método que permite conservar las correlaciones en la medida que los arcos se agrupan, sin embargo, puede que existan mejores maneras de obtener correlaciones equivalentes. Se sugiere estudiar métodos para conseguir estas equivalencias, y verificar el impacto sobre las estimaciones en las distribuciones.

Finalmente, se plantea estudiar la distribución Burr, la cual se dejó fuera de este estudio por motivos que ya fueron tratados. La investigación de *Susilawati et al.* [28] recomienda que esta variable explica de mejor forma los tiempos. Con el fin de dar sustento a esta idea, se analiza una muestra de arcos, y se utiliza esta distribución y la Lognormal para los mismos datos. Dentro de los resultados obtenidos, el promedio de ajuste para la Lognormal es de un 64.4, mientras que la Burr alcanza un 72.36. Además, el porcentaje de aceptados es para la primera variable un 58.33 y para la segunda un 79.17, lo cual en cierta medida, confirma el estudio mencionado.

Sin embargo, esta variable no tiene métodos simples para la suma y el que se conoce es engorroso y asume variables i.i.d. Más allá de eso no se tiene registro, lo cual abre una opor-

tunidad de desarrollo. Se plantea estudiar formas de sumar variables Burr con correlaciones y distintas entre sí, lo cual permitiría incluir esta variable como explicación y predicción de los tiempos de viaje.

Bibliografía

[1] Abramowitz, M. & Stegun, I. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, volume 55. Courier Dover Publications, 1964.

[2] Abu-Dayya, A. A. & Beaulieu, N. C. Comparison of methods of computing correlated lognormal sum distributions and outages for digital wireless applications. In *Vehicular Technology Conference, 1994 IEEE 44th*, pages 175–179. IEEE, 1994.

[3] Ahuja, R. K., Magnanti, T. L. & Orlin, J. B. Network flows: theory, algorithms, and applications. 1993.

[4] Beaulieu, N. C. & Qiong, X. An optimal lognormal approximation to lognormal sum distributions. *Vehicular Technology, IEEE Transactions on*, 53(2):479–489, 2004.

[5] Burr, I. W. Cumulative frequency functions. *The Annals of Mathematical Statistics*, 13(2):215–232, 1942.

[6] Cortés, C. E., Gibson, J., Gschwender, A., Munizaga M., & Zúñiga, M. Commercial bus speed diagnosis based on gps-monitored data. *Transportation Research Part C: Emerging Technologies*, 19(4):695–707, 2011.

[7] DeGroot, M. H., Schervish, M. J., Fang, X., Lu, L. & Li, D. *Probability and statistics*, volume 2. Addison-Wesley Reading, MA, 4th edition edition, 1986.

[8] Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[9] Echeverría, A. R. Evaluación de tiempos de respuesta para el cuerpo de bomberos de santiago: nuevo modelo de ruteo, validación y métricas de desempeño del sistema actual. *Dpto. de Ingeniería Industrial, Universidad de Chile*, 2013.

[10] Eisele, W. L. & Rilett, L. R. Estimating corridor travel time mean, variance, and covariance with intelligent transportation systems link travel time data. In *National Research Council (US). Transportation Research Board. Meeting (81st: 2002: Washington, DC). Preprint CD-ROM*, 2002.

[11] Elmasri, R. & Navathe, S. B. *Fundamentos de sistemas de bases de datos*. Addison-Wesley, 5ta edición edition, 2007.

[12] Faouzi, N. E. & Maurin, M. Reliability metrics for path travel time under log-normal distribution. In *Proceedings of the 3rd International Symposium on Transportation Network Reliability*, 2007.

[13] Fenton, L. F. The sum of log-normal probability distributions in scatter transmission systems. *Communications Systems, IRE Transactions on*, 8(1):57–67, 1960.

[14] Herman, R. & Lam, T. Trip time characteristics of journeys to and from work. In *Transportation and Traffic Theory, Proceedings*, volume 6, 1974.

[15] Knuth, D. E. Fundamental algorithms, the art of computer programming, 1973.

[16] Kortschak, D. & Albrecher, H. An asymptotic expansion for the tail of compound sums of burr distributed random variables. *Statistics & probability letters*, 80(7):612–620, 2010.

[17] Markowitz, H. Portfolio selection*. *The journal of finance*, 7(1):77–91, 1952.

[18] Mehta, N. B., Wu, J., Molisch, A. & Zhang, J. Approximating a sum of random variables with a lognormal. *Wireless Communications, IEEE Transactions on*, 6(7):2690–2699, 2007.

[19] Mendenhall, W., Scheaffer, R. L., Wackerly, D. D., De la Fuente Pantoja, A. & Verbeeck, D. V. *Estadística matemática con aplicaciones*. Grupo Editorial Iberoamericana California, California, 1986.

[20] Newson, P. & Krumm, J. Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 336–343. ACM, 2009.

[21] Osses, M. & Fernandez, R. Transport and air quality in santiago, chile. *Advances in city transport: Case studies*, 2004.

[22] Polus, A. A study of travel time and reliability on arterial routes. *Transportation*, 8(2):141–151, 1979.

[23] Richardson, A. J. & Taylor, M. A. Travel time variability on commuter journeys. *High Speed Ground Transportation Journal*, 12(1), 1978.

[24] Schleher, D. Generalized gram-charlier series with application to the sum of log-normal variates (corresp.). *Information Theory, IEEE Transactions on*, 23(2):275–280, 1977.

[25] Schwartz, S. C. & Yeh, Y. On the distribution function and moments of power sums with lognormal components. *Bell Syst. Tech. J*, 61(7):1441–1462, 1982.

[26] Slimane, S. B. Bounds on the distribution of a sum of independent lognormal random variables. *Communications, IEEE Transactions on*, 49(6):975–978, 2001.

[27] Stuber, G. L. *Principles of mobile communication*. Springer Science Business Media, 2011.

[28] Taylor, M. A. & Somenahalli, S. V. Distribution of variability in travel times on urban roads - a longitudinal study. 2010.

[29] Viterbi, A. J. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269, 1967.

[30] Wang, J., Tsang, W. W. & Marsaglia, G. Evaluating kolmogorov's distribution. *Journal of Statistical Software*, 8(18), 2003.

[31] Wardrop, J. G. Road paper. some theoretical aspects of road traffic research. In *ICE Proceedings: Engineering Divisions*, volume 1, pages 325–362. Ice Virtual Library, 1952.

[32] White, C. E., Bernstein, D. & Kornhauser, A. L. Some map matching algorithms for personal navigation assistants. *Transportation Research Part C: Emerging Technologies*, 8(1):91–108, 2000.

Anexos

Estructura de la librería desarrollada

Arco

- `public Arco(int id, int[] nodo, double[][] coord, Arco[] arco)`
  Constructor de un objeto Arco
  @param `id`: identificador del arco
  @param `nodo`: arreglo de dos enteros que representan nodos
  @param `coord`: arreglo con coordenadas de los nodos
  @param `arco`: arreglo de arcos que es posible alcanzar con este arco

- `public cambiarArcos(Arco[] arco)`
  Cambia el arreglo de arcos que definen los adyacentes
  @param `arco`: arreglo de arcos nuevo

- `public Arco[] getArcos()`
  Obtener los arcos adyacentes
  @return arreglo de arcos

Grafo

- `public Grafo(double minlat, double maxlat)`
  Constructor del objecto Grafo
  @param `minlat`: latitud minima tomando en cuenta los extremos donde esta contenido el grafo
  @param `maxlat`: latitud maxima del grafo
- `public void add(Arco arco)`
  Agrega un arco al grafo
  @param `arco`: objecto arco que se agregara
- `private int findList(int id0, int id1, double latitud)`
  Realiza una busqueda binaria sobre la particion del grafo para encontrar el conjunto que se identifica con la latitud dada
  @param `id0`: id del segmento con que se inicia la participacion

@param `id1`: id del ultimo segmento de la particion
@param `latitud`: latitud de referencia
@return el id del segmento buscado

- `public List get(double latitud)`
  Encuentra el segmento idenficado con una latitud dada
  @param `latitud`: latitud del grafo
  @return lista con los arcos de un segmento dado
- `public Arco contains(int arco)` Busca y devuelve un arco de id dado
  @param `arco`: id del arco que se esta buscando
  @return arco de id dado

Conexión

- `public Conexion(Connection c)`
  Constructor, recibe una conexion a una BD
  @param `c`: connection de la BD
- `public void ejecutarSQL(String sql, String archivo)`
  `throws IOException, SQLException`
  Ejecuta codigo SQL y el resultado lo escribe en un archivo
  @param `archivo`: archivo csv de salida
  @param `sql`: codigo sql a ejecutar
- `public void addGps(String archivo, String codigo)`
  `throws IOException, SQLException`
  Recibe un archivo con datos GPS y el codigo del recorrido que se quiere cargar. Este codigo puede ser null, lo que significa que se cargan todos los datos
  @param `archivo`: archivo de entrada de datos
  @param `codigo`: codigo del recorrido a cargar
- `public void actualizarDatos(String nodo, String arco)`
  `throws IOException, SQLException`
  Actualiza los datos del grafo en la base de datos. Si el arco y/o nodo ya existen, entonces reemplaza con los nuevos valores
  @param `nodo`: archivo que contiene los nodos del grafo
  @param `arco`: archivo que contiene los arcos del grafo
- `public double distancia(double lat1, double long1,`
  `double lat2, double long2)`
  Calcula la distancia entre dos puntos
  @param `lat1`: latitud del punto 1
  @param `long1`: longitud del punto 1
  @param `lat2`: latitud del punto 2
  @param `long2`: longitud del punto 2
  @return valor de distancia
- `public double[][] encontrarArcos(double latitud, double longitud,`
  `int[] arcos, double dist)`
  `throws SQLException`
  Para un conjunto de arcos, calcula la distancia a un punto GPS

@param `latitud`: latitud del punto GPS
@param `longitud`: longitud del punto GPS
@param `arcos`: conjunto de arcos
@param `dist`: distancia maxima entre el punto GPS y un arco
@return arreglo que en cada fila contiene id del arco, distancia a este, proyeccion del punto GPS en el arco, id del nodo de inicio y distancia desde la proyeccion y el nodo de inicio

- `private double[][] encontrarArcos(double latitud, double longitud, double dist)`
  Proyecta un punto GSP en arcos de un subgrafo almacenado en la estructura grafo de la clase. Utiliza como criterio una distancia maxima. Retorna todos los arcos donde se puede proyectar, en orden segun la distancia desde el punto GPS al arco.
  @param `latitud`: latitud del punto GPS
  @param `longitud`: longitud del punto GPS
  @param `dist`: distancia maxima del arco a proyectar
  @return Arreglo de arreglos double[], cada uno con los siguientes datos en orden: id del arco, distancia a ese arco, latitud y longitud proyectada, el id del nodo donde comienza el arco y la distancia del nodo inicio a la proyeccion al arco
- `public Object[] ArmarRecorrido(String codigo) throws SQLException`
  Define el recorrido para una linea determinada
  @param `codigo`: linea del recorrido
  @return retorna un arreglo de objetos. En la posición [0] esta el camino final seleccionado por el algoritmo, en la posicion [1] un verdadero o falso y la [2] los puntos gps que dieron origen a este camino. Estos resultados son para el sentido 1. El sentido 2 tiene las mismas entregas en las posiciones [3], [4] y [5]
- `private Object[] seleccionarPuntos(Object[] set) throws SQLException`
  Recibe un conjunto de puntos gps ordenados y encuentra el camino que siguieron
  @param `set`: Un arreglo de varios caminos (al menos 150) que realizo un recorrido. Cada fila es un conjunto de puntos ordenados.
  @return Un arreglo de objetos. En [0] se entrega un arreglo de arcos que fueron seleccionados, en [1] un validador de coincidencias y en [3] todos los puntos gps que se usaron para encontrar el camino
- `private Object[] HMM(double[][] puntos)`
  Encuentra el camino mas probable de arcos para el set de observaciones usando Hidden Markov Model
  @param `puntos`: set de datos latitud, longitud de las observaciones, ordenadas temporalmente
  @return Devuelve un arreglo de objectos. En [0], el camino de arcos para la muestra particular y en [1] un check si logro conectar todos los puntos
- `private Object[] Viterbi(int[] obs, int[] states, Hashtable<Integer, Hashtable<Integer, Double» prob_trans, Hashtable<Integer, Hashtable<Integer, Double» prob_em)`
  Resuelve una cadena de Markov oculta usando el algoritmo de Viterbi
  @param `obs`: conjunto de observaciones de la cadena

@param `states`: conjunto de estados
@param `prob_trans`: probabilidades de transicion entre estados
@param `prob_em`: probabilidades que relacionan una observacion con un estado de la cadena
@return devuelve un Objecto con los arcos (estados) mas probables y un boolean que indica si fue posible resolver la cadena

- `private Object[] ShortestPath(int a1, int a2)`
  Encuentra el camino minimo entre n1 y n2
  @param `n1`: id arco inicio
  @param `n2`: id arco fin
  @return array de object con dos valores, primero un array con los ids de arcos entre un punto y otro y una dist total entre un punto y otro. Esta distancia puede ser -1 si no encuentra un camino.
- `public void Proyectar(String codigo, int sentido, double dist, String date) throws SQLException`
  Proyecta los puntos GPS y calcula las velocidades asociadas a estos puntos
  @param `codigo`: codigo del recorrido
  @param `sentido`: sentido de viaje (1 o 2)
  @param `dist`: distancia maxima de proyeccion de un punto
  @param `date`: dia que se va a cargar (YYYYMMDD), puede ser null lo que significa que se cargan todos los dias
- `private Object[][] Proyectar(ResultSet rs, Object[] p0, Object[] p1, int[] arcos, double[][] nodos, String[] recorrido, double dist) throws SQLException`
  Corresponde la una iteracion en la proyeccion de los puntos GPS. Inserta los datos en la tabla trayecto de la base de datos
  @param `rs`: conjuntos de valores extraidos que seran proyectados
  @param `p0`: set de informacion del primer punto donde se proyecto la secuencia
  @param `p1`: set de informacion del segundo punto donde se proyecto la secuencia
  @param `arcos`: conjunto de todos los arcos del camino @param `nodos`: nodos del camino
  @param `recorrido`: arreglo con el codigo del servicio y el sentido que se esta proyectado
  @param `dist`: distancia maxima para proyectar
  @return informacion del siguiente conjunto de datos para proyectar
- `public float[][] KStest(int[] arcos) throws SQLException`
  Para un conjunto de arcos, calcula el estadistico Kolmogorov-Smirnov usando todas las combinaciones arco, hora
  @param `arcos`: arreglo de id de arcos
  @return matriz de estadisticos
- `private double KStest(int arco1, int hora1, int arco2, int hora2) throws SQLException`
  Calcula el estadistico de Kolmogorov-Smirnov para un par (arco,hora)
  @param `arco1`: id del primer arco
  @param `hora1`: hora del primer arco @param `arco2`: id del segundo arco

@param `hora2`: hora del segundo arco
@return estadistico

- `public double KSTestLogN(int arco, int hora)`
  `throws SQLException`
  Realiza el test de KS para un par (arco,hora), comparandolo con la Lognormal que se puede estimar a traves de los datos
  @param `arco`: id del arco a evaluar
  @param `hora`: hora del dia seleccionada
  @return valor entre 0 y 1 del estadistico
- `public double KSDistribution(double n, double d)`
  Calcula el valor de la distribucion Kolmogorov-Smirnov
  @param `n`: numero de datos usados para el test
  @param `d`: distancia maxima entre una distribucion y otra
  @return valor de la distribucion para estos parametros