



# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE CIENCIAS**

### **MODELOS INTEGRADOS DE ÓPTIMIZACIÓN DE TRANSPORTE PÚBLICO**

### **HEURISTICAS PARA EL PROBLEMA INTEGRADO DE CALENDARIZACIÓN Y ENRUTAMIENTO DE PASAJEROS**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO  
MATEMÁTICO**

**ESTEBAN JAVIER GUACHAMIN SÁNCHEZ**

[esteban.guachamin@epn.edu.ec](mailto:esteban.guachamin@epn.edu.ec)

**DIRECTOR: DRA. MARÍA FERNANDA SALAZAR MONTENEGRO**

[fernanda.salazar@epn.edu.ec](mailto:fernanda.salazar@epn.edu.ec)

**DMQ, MARZO 2023**

## **CERTIFICACIONES**

Yo, ESTEBAN JAVIER GUACHAMIN SÁNCHEZ, declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

---

Esteban Javier Guachamin Sánchez

Certifico que el presente trabajo de integración curricular fue desarrollado por Esteban Javier Guachamin Sánchez, bajo mi supervisión.

---

Dra. María Fernanda Salazar Montenegro  
**DIRECTOR**

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el(los) producto(s) resultante(s) del mismo, es(son) público(s) y estará(n) a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Esteban Javier Guachamin Sánchez

Dra. María Fernanda Salazar Montenegro

**Agradecimientos:**

Quiero agradecer a Dios por todas las bendiciones que me da cada día, en especial por mi familia.

Le doy gracias a mi madre, a mis hermanos, a mis abuelitos, por su amor y toda la paciencia durante mi vida universitaria.

A la Dra. Fernanda Salazar, que me acompañó durante todo el proceso de este trabajo hasta el final, le doy gracias por los conocimientos impartidos y la paciencia.

Y a mi hermosa prometida Andrea que es uno de los motores más importantes de mi vida.

**Dedicatoria:**

Quiero dedicar el presente trabajo a mis hermanos Tania, Cristina, Kevin, Alan, pero sobre todo a mi madre Raquel por su esfuerzo durante de cada día. Les quiero mucho.

## RESUMEN

En el marco de los modelos de optimización para la planificación del transporte público, en este trabajo se consideran dos etapas importantes: Calendarización de Viajes y Enrutamiento de Pasajeros. El problema de Enrutamiento de Pasajeros se refiere a encontrar la cantidad de pasajeros que usan cada una de las diferentes rutas disponibles de la red de transporte dentro de un cierto horizonte de tiempo mientras que la definición del problema general de Calendarización de viajes consiste en determinar los tiempos de salida y llegada de buses a todas las paradas a lo largo de una red de transporte para cumplir diferentes metas como: cumplir con una frecuencia dada, satisfacer los patrones de demanda, maximizar el número de buenos tiempos de transferencia de pasajeros, y minimizar tiempos de espera.

En este trabajo se estudiará el problema de calendarización de viajes de forma individual, haciendo énfasis en el enfoque de solución dado por el modelo de programación entera PESP (Periodic Event Scheduling Problem), y se realizará un análisis más profundo del modelo integrado de calendarización de viajes y ruteo Tim-Pass (Integrated Timetabling and Passenger Routing Problem), ambos modelos son ampliamente conocidos en la literatura. La parte central del trabajo se refiere a analizar opciones que permitan obtener una solución al modelo integrado, que es un modelo entero. Para ello se abordarán dos heurísticas conocidas como: LB (Cota Inferior) y UB (Cota Superior), centrando la atención en la LB cuyo enfoque está orientado a la elección de un subconjunto de pares OD (origen-destino).

**Palabras clave:** Heurísticas, Cota inferior, Calendarización, Enrutamiento, Preprocesamiento.

## **ABSTRACT**

In this work we are interested in optimization models for public transport planning. Specifically, two important stages are considered: Timetabling and Passenger Routing. The latter refers to finding the number of passengers using each of the different routes available in the transportation network within a certain time horizon, while the definition of the Timetabling problem consists of determining the departure and arrival times of buses at all stops along a transportation network to meet different goals such as: meeting a given frequency, satisfying demand patterns, maximizing the number of good passenger transfer times, and minimizing timeouts.

In this work, the timetabling problem will be studied individually, emphasizing the solution approach given by the PESP (Periodic Event Scheduling Problem), whereas a deeper analysis of the integrated model for Timetabling and Passenger Routing will be carried out by the study of Tim-Pass (Integrated Timetabling and Passenger Routing Problem). Both models are well known in the literature.

The core of the present study is the analysis of methods that allow obtaining a solution for the integrated model, which is an integer programming model. To do this, two heuristics known as: LB (Lower Bound) and UB (Upper Bound) will be addressed, focusing attention on the LB to evaluate it with a preprocessing algorithm whose approach is oriented to the choice of a subset of OD pairs (origin- destination).

**Keywords:** Heuristics, Lower Bound, Timatabling, Routing, Preprocessing.

---

# Índice general

---

<b>1. Introducción</b>	<b>1</b>
<b>2. Marco Teórico</b>	<b>4</b>
2.1. Preliminares . . . . .	4
2.2. Calendarización de viajes: Modelo PESP . . . . .	6
2.3. Calendarización de Viajes y Ruteo de Pasajeros: Modelo Tim-Pass . . . . .	12
2.4. Reducción del tamaño del problema . . . . .	14
2.4.1. Heurística LB . . . . .	16
2.4.2. Heurística UB . . . . .	18
2.4.3. Algoritmo de preprocesamiento . . . . .	20
2.4.4. Selección de pares OD . . . . .	22
<b>3. Resultados computacionales</b>	<b>23</b>
3.1. Descripción de las instancias . . . . .	23
3.1.1. Instancia Pequeña 1 . . . . .	23
3.1.2. Instancia Pequeña 2 . . . . .	24
3.1.3. Instancia Grid 5x5 . . . . .	24
3.2. Resultados y análisis: . . . . .	25
3.2.1. Resultado Ilustrativo: Instancia Juguete . . . . .	25



3.2.2. Resultados Instancia Pequeña 2 . . . . .	26
3.2.3. Resultados Instancia Grid 5x5 . . . . .	27
<b>4. Conclusiones y Recomendaciones</b>	<b>29</b>
<b>Bibliografía</b>	<b>31</b>

---

## Índice de figuras

---

2.1. Ejemplo de <b>PTN</b> con 5 paradas y 2 líneas . . . . .	6
2.2. Ejemplo de <b>EAN</b> asociada a la <b>PTN</b> de 2.1 . . . . .	8
2.3. Ejemplo de <b>EAN</b> extendida asociada a la <b>EAN</b> de 2.2 . . . . .	10
2.4. <b>Instancia Pequeña 2 [4]</b> . . . . .	14
2.5. Algoritmo de preprocesamiento instancia long-distance [3] . . . . .	22
3.1. Gráfico de la <b>PTN</b> de la instancia <b>Pequeña 1</b> . . . . .	24
3.2. Gráfico de la <b>PTN</b> de la instancia <b>Pequeña 1</b> . . . . .	24
3.3. Resultados: Instancia Juguete . . . . .	25
3.4. Resultados: Instancias que alcanzan el óptimo . . . . .	26
3.5. Resultados que no alcanzan el óptimo: . . . . .	27
3.6. Resultados que no alcanzan el óptimo: . . . . .	28
3.7. Resultados: Instancias que alcanzan el óptimo . . . . .	28

# Capítulo 1

---

## Introducción

---

Varios problemas que enfrenta el sector público están relacionados con problemas de optimización, tales como:: minimizar gastos, minimizar recursos, maximizar la producción, maximizar la cobertura de servicios, entre otras. Por lo que tomar una buena decisión beneficia a la ciudadanía en general. Muchas de las decisiones importantes que tiene el sector público, giran en torno a cómo proponer, proyectar, controlar y gestionar un buen sistema de transporte público que se ajuste a las necesidades de los usuarios que, a medida que pasa el tiempo, van aumentando y que, sin un criterio de control, provoca que la movilidad en los sectores principales no sea buena.

Para que un sistema de transporte público logre generar un impacto positivo se deben tomar en cuenta los factores más importantes para los principales involucrados dentro del sistema que son los usuarios, operarios y, en el sentido financiero, los propietarios de las empresas de transporte, que, en el día a día tienen problemas que aparentan ser distintos (debido al papel que juegan dentro del sistema de transporte público), pero que en realidad tienen mucho que ver; por ejemplo, como se escucha en el día a día a los operarios de los colectivos de Quito: "Cómo quiere que me apure si adelante ya tengo a toda la flota?", o también el " ¡Ya no hay dónde!" por parte de los pasajeros molestos que han esperado tiempos extremadamente largos por una unidad que excede la capacidad permitida, quiénes incluso luego tendrán que hacer transbordos a uni-

dades con las mismas o incluso peores características, sin mencionar a los propietarios que seguramente tienen el problema de cubrir los gastos de un viaje vacío al inicio de la jornada desde los depósitos hasta las paradas de inicio.

Como se menciona en-[2], el proceso de planificación que considera todos los problemas de operación de transporte público (incluso los mencionados anteriormente) se denomina Problema de Planificación de la Red de Tránsito (**TNP** por sus siglas en inglés). Debido a la complejidad del **TNP** comúnmente se dividen en algunos subproblemas como los presentados por [8] que centra la atención en los problemas de planificación de líneas, enrutamiento de pasajeros, calendarización de viajes y asignación de vehículos :

- [1] propone que el problema general de **Planificación de líneas** consiste en encontrar un conjunto de líneas con sus respectivas frecuencias, con el objetivo de cubrir la demanda del transporte garantizando un cierto nivel de servicio y minimizando los costos de operación.
- En [4] se considera el problema general de **Enrutamiento de Pasajeros** como encontrar la cantidad de pasajeros que usan cada una de las diferentes rutas disponibles de la red de transporte dentro de un cierto horizonte de tiempo.
- La definición del problema general de **Calendarización de viajes** presentado en [2] consiste en determinar los tiempos de salida y llegada de buses a todas las paradas a lo largo de una red de transporte para cumplir diferentes metas como: cumplir con una frecuencia dada, satisfacer los patrones de demanda, tener buenos tiempos de transferencia de pasajeros, y minimizar tiempos de espera.
- [2] define el problema de **Asignación de Vehículos** como determinar la asignación de viajes de los vehículos para cubrir todos los viajes planeados tal que los costos de operación sobre los usos de los vehículos sea minimizado.

Tratar de resolver estos problemas por separado puede resultar muy beneficioso para ciertas partes del sistema de transporte público pero debi-

do a que en el momento en el que se resuelve un problema por separado se ignora por completo las demás necesidades del sistema, puede ocasionar que las piezas ignoradas vayan mucho peor que a como iban antes de la solución de un problema aparentemente independiente de los otros.

Por esta razón en investigaciones como [8], [7] han logrado desarrollar recursos como [5] para abordar de manera integrada los problemas presentados que, aunque suene prometedor para el transporte público incluso con los avances tecnológicos, no son fáciles de resolver, como se menciona en los resultados computacionales de [10] para el problema de calendarización de viajes y asignación vehículos con 8 horas de cálculo en el súper computador **Quinde** del Centro de Modelización Matemática se pudieron resolver hasta el óptimo.

Como se define en [9], un método heurístico es un procedimiento para resolver un problema matemático bien definido mediante un enfoque intuitivo en el que la estructura general puede interpretarse inteligentemente para obtener una solución razonable. El presente trabajo centra la atención en el estudio del problema integrado de **Calendarización de viajes y Enrutamiento de Pasajeros**, principalmente en cómo disminuir inteligentemente el tamaño del problema fijando el interés en descartar variables asociadas a actividades que nunca serán utilizadas y dando prioridad a trayectos de origen y destino de interés.

# Capítulo 2

---

## Marco Teórico

---

### 2.1. Preliminares

Para presentar el enfoque en el que se basará este trabajo es necesario definir nociones básicas de teoría de grafos y notación de transporte público que son adaptaciones de [8]:

**Definición 1.** Un grafo no dirigido  $G = (V, E)$  consiste en un conjunto de nodos o vértices  $V$  y un conjunto de aristas  $E \subset \{\{i, j\} : i, j \in V\}$ . Dada una arista  $e = \{i, j\} \in E$ , se dice que  $e$  conecta al nodo  $i$  con el nodo  $j$ .

**Definición 2.** Un grafo dirigido  $G = (V, A)$  consiste en un conjunto de nodos  $V$  y de un conjunto de arcos  $A \subset V \times V$ . Dado un arco  $a = (i, j) \in A$ , se dice que  $a$  conecta el nodo  $i$  con el nodo  $j$ .

**Definición 3.** Un camino en un grafo no dirigido consiste en una lista de nodos y aristas  $P = (v_1, e_1, v_2, e_2, \dots, e_{n-1}, v_n)$  tal que las aristas satisfacen que  $e_i = \{v_i, v_{i+1}\}$  para todo  $i \in \{1, \dots, n-1\}$ .

El camino también puede definirse solo por aristas:  $P = (e_1, \dots, e_{n-1})$ , nótese que se puede escribir  $v \in P$  y  $e \in P$  para nodos o aristas que forman parte del camino  $P$ .

Este concepto puede extenderse para grafos dirigidos, cambiando las aristas por arcos.

**Definición 4.** Un camino  $P$  que no contiene nodos repetidos se llama camino simple.

**Definición 5.** Dado un grafo  $G = (V, E)$  no dirigido y un camino  $P$  en  $G$ , sea  $len$  una función de costos definida como  $len : E \rightarrow \mathbb{R}$ . La longitud o costo de un camino se define como:

$$len(P) = \sum_{e \in P} len(e)$$

La definición es análoga para grafos dirigidos.

**Definición 6.** Sea un grafo dirigido  $G = (V, A)$ . Un camino  $s - t$  es un camino  $P = (s, a_1, v_2, \dots, a_{n-1}, t)$  que empieza en el nodo  $s$  y termina en el nodo  $t$ .

El camino más corto de  $s$  a  $t$ , es un camino  $s-t$  que tiene longitud mínima.

**Definición 7.** Una Red de Transporte Público (**PTN**: Public Transportation Network). Es un grafo  $G = (V, E)$  cuyos nodos  $V$  corresponden a las estaciones o paradas y cuyas aristas  $E$  corresponden a las conexiones directas de transporte entre paradas. La forma que tenga la PTN dirá si se usan direcciones en las aristas o no.

Para ilustrar esta definición, ver el ejemplo 2.1 de una **PTN** con 5 paradas y 2 líneas:  $l_1$  verde y  $l_2$  morado. **Nota:** Este ejemplo es ilustrativo, por ello no están considerados los de retornos de las líneas .

**Definición 8.** Sea una Red de Transporte Público  $G = (V, E)$ , la matriz  $C = (c_{u,v})_{u,v \in V}$  es una matriz tal que  $c_{u,v} \in \mathbb{Z}$  indica la cantidad de pasajeros que viajan desde la estación  $u$  hasta la estación  $v$  dentro de un horizonte de tiempo.

A esta matriz se la conoce como matriz origen-destino o matriz-OD y se escribe  $(u, v) \in C$  para indicar que  $C_{u,v} > 0$

**Definición 9.** Una línea  $l$ , es un camino simple dentro de una **PTN**, la cual es operada de principio a fin por un mismo vehículo,  $\mathcal{L}$  es el con-

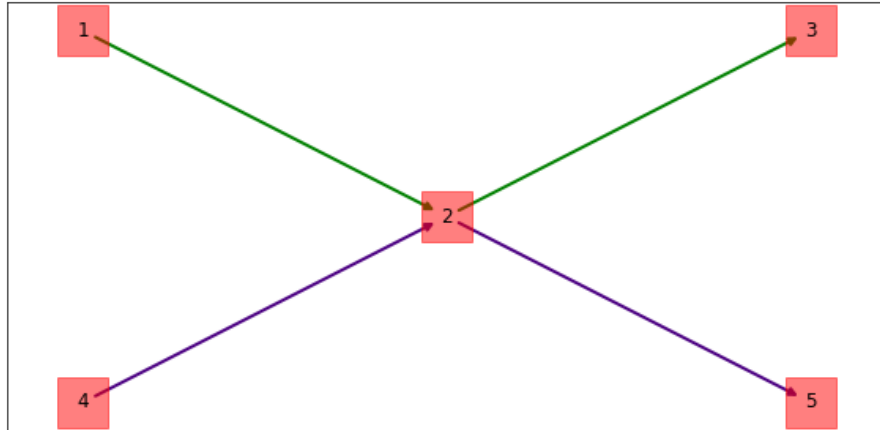


Figura 2.1: Ejemplo de **PTN** con 5 paradas y 2 líneas

junto de líneas que son operadas dentro de la **PTN**.

**Definición 10.** Sea un conjunto de líneas  $\mathcal{L}$ . Se llamará  $f_l$  a la frecuencia de una línea  $l$ , que indicará el número de viajes que se realizan en un periodo de planificación.

Note que cuando  $f_l \geq 1$ , por ejemplo  $f_l = 2$ , se puede trabajar con dos copias  $f_{l_1}, f_{l_2}$  de la misma línea  $f_l$ , pero cada una con frecuencia igual a 1. Esto permitirá que más adelante se puedan considerar las frecuencias dentro del modelamiento.

## 2.2. Calendarización de viajes: Modelo PESP

El problema de calendarización de viajes es el tercer problema que se aborda en [8], consiste en encontrar el mejor calendario para los eventos de llegada y salida de vehículos de las estaciones dentro de un periodo de tiempo. Sin embargo, para modelar este problema es necesario generar una nueva red que tome en cuenta la información conjunta de: llegada y salida de las paradas, las líneas las cuales hacen posible la visita de los vehículos a las paradas, las actividades de espera, transbordo y viaje que se realizarán dentro del sistema.

Note que al ser esta la tercera etapa, se considerará que los resultados de



las anteriores son datos conocidos, es decir las líneas y el enrutamiento son datos. Sin embargo, para considerar el ruteo de pasajeros fijo es necesario resolver el problema de enrutamiento con la información de la matriz de origen destino. En esta sección se mostrará la manera de resolver el problema de ruteo.

**Definición 11.** Dada una PTN  $G = (V, E)$ . La **Red Evento Actividad (EAN)** asociada a la **PTN** es un grafo dirigido  $\mathcal{N}^0 = (\mathcal{E}^0, \mathcal{A}^0)$  donde  $\mathcal{E}^0$  es el conjunto de nodos llamado eventos y  $\mathcal{A}^0$  el conjunto de aristas llamado actividades.

Los nodos van a representar los eventos de llegada y salida de vehículos de las paradas y las actividades representan el viaje de los vehículos, tiempo de espera de vehículos y transferencia de pasajeros.

Para una Red de Transporte Público  $(V, E)$  y un conjunto de líneas  $\mathcal{L}$  la Red Evento Actividad se construye de la siguiente manera:

$$\mathcal{E}_{arr} = \{(v, arr, l) : v \in l \cap V, l \in \mathcal{L}\}$$

El conjunto de nodos de llegada a la estación  $v \in l \cap V$  de la línea  $l \in \mathcal{L}$ ,

$$\mathcal{E}_{dep} = \{(v, dep, l) : v \in l \cap V, l \in \mathcal{L}\}$$

El conjunto de nodos de salida a la estación  $v \in l \cap V$  de la línea  $l \in \mathcal{L}$ ,

$$\mathcal{A}_{drive} = \{((v_1, dep, l), (v_2, arr, l)) : e = (v_1, v_2) \in l \cap E, l \in \mathcal{L}\}$$

El conjunto aristas de viaje: desde el nodo de salida  $(v_1, dep, l)$  al nodo llegada  $(v_2, dep, l)$  para los cuales  $v_1, v_2 \in l \cap E, l \in \mathcal{L}$ .

$$\mathcal{A}_{wait} = \{((v, arr, l), (v, dep, l)) : v \in l \cap V, l \in \mathcal{L}\}$$

El conjunto aristas de espera: desde el nodo de llegada  $(v, arr, l)$  al nodo salida  $(v, dep, l)$  para los cuales  $v \in l \cap E, l \in \mathcal{L}$ .

$$\mathcal{A}_{trans} = \{((v, arr, l_1), (v, dep, l_2)) : v \in l_1 \cap l_2 \cap V, l_1, l_2 \in \mathcal{L}, l_1 \neq l_2\}$$

El conjunto aristas de transbordo: desde el nodo de llegada  $(v, arr, l_1)$  al nodo salida  $(v, dep, l_2)$  para los cuales  $v \in l_1 \cap l_2 \cap V$ ,  $l_1, l_2 \in \mathcal{L}$ ,  $l_1 \neq l_2$  (Para los cuales tengan paradas comunes y no sean las mismas líneas).

Finalmente se definen los conjuntos que general la **EAN**  $\mathcal{N}^0 = (\mathcal{E}^0, \mathcal{A}^0)$ :

$$\mathcal{A}^0 = \mathcal{A}_{drive} \cup \mathcal{A}_{wait} \cup \mathcal{A}_{trans}$$

$$\mathcal{E}^0 = \mathcal{E}_{arr} \cup \mathcal{E}_{dep}$$

Para cada actividad  $a \in \mathcal{A}^0$ , las cotas superiores e inferiores de duración están dadas por  $L_a, U_a \in \mathbb{N}$  con  $0 \leq L_a \leq U_a$ , las cuales jugarán el papel de función de costos de los arcos asociados a la **EAN**.

Para ilustrar la **EAN**, ver 2.2.

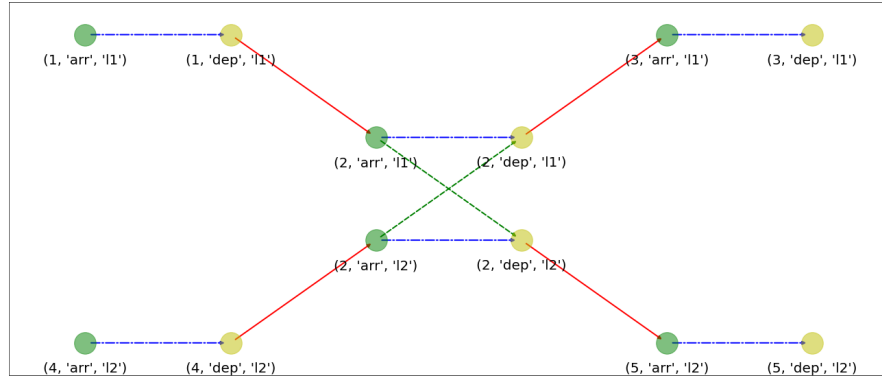


Figura 2.2: Ejemplo de **EAN** asociada a la **PTN** de 2.1

**Definición 12.** Sea  $\mathcal{N}^0 = (\mathcal{E}^0, \mathcal{A}^0)$  una EAN con cotas  $L_a, U_a$  para las actividades  $a \in \mathcal{A}^0$  y sea  $T \in \mathbb{N}$  un periodo de planificación. Se dice que  $\pi = (\pi_i)_{i \in \mathcal{E}^0} \in \{0, \dots, T-1\}^{|\mathcal{E}^0|}$  es un calendario periódico factible que asigna tiempos periódicos a los eventos  $i \in \mathcal{E}^0$  tal que la duración de la actividad  $a \in \mathcal{A}^0$  se encuentre dentro de las cotas  $[L_a, U_a]$ , es decir se satisface que:

$$L_a \leq (\pi_j - \pi_i - L_a) \bmod T + L_a \leq U_a \quad \forall a = (i, j) \in \mathcal{A}^0$$

**Definición 13.** La Red Evento Actividad Extendida  $\mathcal{N} = (\mathcal{E}, \mathcal{A})$  se deriva de la Red Evento Actividad  $\mathcal{N}^0 = (\mathcal{E}^0, \mathcal{A}^0)$  para una Red de Transporte Público  $G = (V, E)$  añadiendo eventos de fuente y destino para todas las paradas, también las correspondientes actividades que conectan los eventos fuente y destino con las salidas y llegadas a las respectivas paradas:

$$\mathcal{E}_{source} = \{(v, source) : v \in V\},$$

Conjunto de nodos fuente de las paradas  $v \in V$ ,

$$\mathcal{E}_{target} = \{(v, target) : v \in V\},$$

Conjunto de nodos destino de las paradas  $v \in V$ ,

$$\mathcal{A}_{to} = \{((v, source), (v, dep, l)) : v \in l \cap V, l \in \mathcal{L}\},$$

Conjunto de aristas que van desde el nodo  $(v, source)$  al nodo  $(v, dep, l)$ , para todas las paradas  $v \in l \cap V, l \in \mathcal{L}$ ,

$$\mathcal{A}_{from} = \{((v, arr, l), (v, target)) : v \in l \cap V, l \in \mathcal{L}\},$$

Conjunto de aristas que van desde el nodo  $(v, arr, l)$  al nodo  $(v, target)$ , para todas las paradas  $v \in l \cap V, l \in \mathcal{L}$ . Finalmente, se definen los conjuntos que definen la **EAN** extendida:

$$\mathcal{E} = \mathcal{E}^0 \cup \mathcal{E}_{source} \cup \mathcal{E}_{target} \quad con$$

$$\mathcal{A} = \mathcal{A}^0 \cup \mathcal{A}_{to} \cup \mathcal{A}_{from} \quad con$$

Las cotas superiores e inferiores de las actividades auxiliares  $\mathcal{A}_{aux} = \mathcal{A}_{to} \cup \mathcal{A}_{from}$  se establecen en cero, es decir, para  $a \in \mathcal{A}_{aux}$  se tiene que  $L_a = U_a = 0$ .

Para ilustrar la **EAN** extendida, ver [2.2](#).

Como se mencionó anteriormente, para considerar el problema de la fase

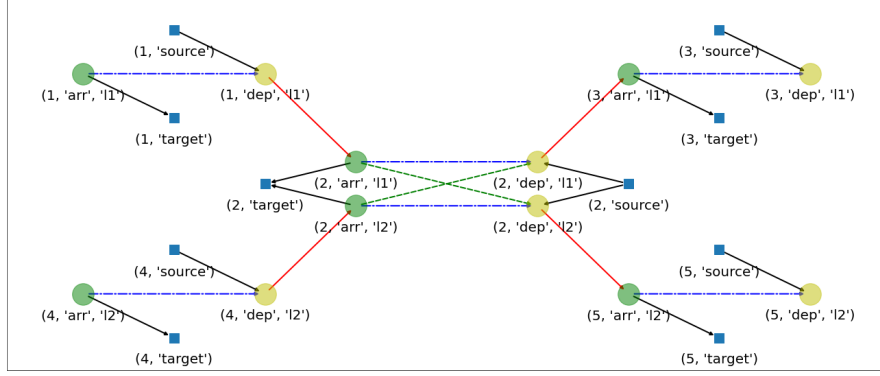


Figura 2.3: Ejemplo de **EAN** extendida asociada a la **EAN** de 2.2

anterior como fijo es necesario resolver el problema de enrutamiento.

El Problema de enrutamiento de pasajeros consiste en conocer el número de pasajeros que se encuentran haciendo uso de las actividades de manera que se encuentren direccionados desde su estación de origen a su estación de destino. Para resolver este problema considere, las siguientes dos definiciones:

**Definición 14.** Sea una **PTN**  $G = (V, E)$ , una **EAN**  $\mathcal{N}^0 = (\mathcal{E}^0, \mathcal{A}^0)$  y una matriz OD  $C$ . Un **Ruteo de pasajeros** es un conjunto  $\mathcal{R}$  formado por todos los caminos que contienen un camino desde  $u \in V$  hasta  $v \in V$  en la **EAN** extendida desde  $(u, source)$  hasta  $(v, target)$ , estos caminos son llamados Rutas del Pasajero.

**Definición 15.** Dada una **EAN**  $\mathcal{N}^0 = (\mathcal{E}^0, \mathcal{A}^0)$  y un ruteo de pasajeros  $\mathcal{R}$ , la carga de los pasajeros inducida por  $\mathcal{R}$  sobre  $\mathcal{N}^0$  es  $w = (w_a)_{a \in \mathcal{A}^0}$  con  $w_a \geq 0, a \in \mathcal{A}^0$  que indica el número de pasajeros  $w_a$  usando la actividad  $a \in \mathcal{A}^0$  dentro de sus rutas, la tupla está dada por:

$$|\{P \in \mathcal{R} : a \in P\}| = w_a, \quad \forall a \in \mathcal{A}^0$$

**Definición 16.** Dado  $\mathcal{N}^0 = (\mathcal{E}^0, \mathcal{A}^0)$  una Red Evento Actividad con cotas  $L_a, U_a, a \in \mathcal{A}^0$ , pesos de pasajeros  $w = (w_a)_{a \in \mathcal{A}^0}$  y un periodo de planifica-

ción  $T$ . Un calendario periódico factible  $\pi = (\pi_i)_{i \in \mathcal{E}^0} \in \{0, \dots, T-1\}^{|\mathcal{E}^0|}$  asigna tiempos periódicos  $\pi_i$  a los eventos  $i \in \mathcal{E}^0$  tal que la duración de la actividad  $a \in \mathcal{A}^0$  se encuentre dentro de  $[L_a, U_a]$ , es decir que  $\pi$  cumpla que:

$$L_a \leq (\pi_j - \pi_i - L_a) \bmod T + L_a \leq U_a, \quad \forall a = (i, j) \in \mathcal{A}^0$$

**Definición 17.** Dado  $\mathcal{N}^0 = (\mathcal{E}^0, \mathcal{A}^0)$  una Red Evento Actividad con cotas  $L_a, U_a, a \in \mathcal{A}^0$ , pesos de pasajeros  $w = (w_a)_{a \in \mathcal{A}^0}$  y un periodo de planificación  $T$ . El Problema de Calendarización de Eventos Periódicos (**PESP**: Periodical Event Scheduling Problem) consiste en buscar un calendario periódico factible  $\pi = (\pi_i)_{i \in \mathcal{E}^0} \in \{0, \dots, T-1\}^{|\mathcal{E}^0|}$  tal que la duración ponderada

$$\sum_{a=(i,j) \in \mathcal{A}^0} w_a ((\pi_j - \pi_i - L_a) \bmod T + L_a)$$

sea minimizada.

En [3] se propone el modelo **PESP** de la siguiente manera:

$$\min \sum_{a=(i,j) \in \mathcal{A}^0} w_a \cdot (\pi_j - \pi_i + z_a \cdot T)$$

s.t

$$\pi_j - \pi_i + z_a T \geq L_a \quad a = (i, j) \in \mathcal{A}^0$$

$$\pi_j - \pi_i + z_a T \leq U_a \quad a = (i, j) \in \mathcal{A}^0$$

$$\pi_i \in \{0, \dots, T-1\} \quad i \in \mathcal{E}^0$$

$$z_a \in \mathbb{Z} \quad a \in \mathcal{A}^0$$

En donde se usan variables auxiliares  $z_a \in \mathbb{Z}, a \in \mathcal{A}^0$  llamadas parámetros de módulo para las actividades que modelan la periodicidad del horario.

## 2.3. Calendarización de Viajes y Ruteo de Pasajeros: Modelo TimPass

El problema de interés en esta sección es la Calendarización de Viajes y Ruteo de Pasajeros, que integra la segunda y tercera fase de planificación de transporte presentado en [8]. Este problema consiste en encontrar el mejor calendario para los eventos de llegada y salida de vehículos de las estaciones dentro de un periodo de planificación tomando en cuenta las necesidades de los pasajeros de ir de una estación a otra. Para resolver este problema es necesario considerar por dónde se pueden mover los pasajeros, para lo cual es necesario introducir variables de flujo que como se verá más adelante, son las responsables del gran tamaño que tiene este problema.

A diferencia del modelo **PESP** en esta parte lo único que se considerará como dato son las líneas de la **PTN**, las cotas  $L_a, U_a$ , el periodo  $T$  y la matriz OD.

**Definición 18.** Para cada actividad  $a \in \mathcal{A}$  definimos su duración respecto a un calendario  $\pi$  como:

$$d_a(\pi) = \begin{cases} (\pi_j - \pi_i - L_a) \bmod T + L_a & \text{si } a = (i, j) \in \mathcal{A}^0 \\ 0 & \text{si } a = (i, j) \in \mathcal{A}_{aux} \end{cases}$$

Notemos por  $d(\pi) = (d_a(\pi))_{a \in \mathcal{A}}$ ,  $L = (L_a)_{a \in \mathcal{A}}$  y  $U = (U_a)_{a \in \mathcal{A}}$  para abreviar al vector de duraciones, los vectores de las cotas inferior y superior respectivamente.

**Definición 19.** Sea  $\beta = (\beta_a)_{a \in \mathcal{A}}$  un vector de duración de actividades (o una función de costos sobre la **EAN** extendida). Entonces:

- $SP_{u,v}(\beta)$  denota el camino más corto para un OD par  $(u, v)$  desde  $(u, source)$  hasta  $(v, target)$  acorde la duración de las actividades  $\beta_a, a \in \mathcal{A}$
- Y la longitud de un camino  $P$  asociado al vector de duraciones  $\beta$  se

define como:

$$\text{len}(P, \beta) = \sum_{a \in P} \beta_a$$

Se tiene entonces que  $\text{len}(SP_{u,v}(\beta), \beta)$  es la longitud del camino más corto con respecto a la duración de actividades  $\beta$ , además note que  $\text{len}(SP_{u,v}(\beta), \gamma)$  describe la longitud del camino más corto  $SP_{u,v}(\beta)$  (con respecto a  $\beta$ ) pero evaluado con otro vector de duraciones  $\gamma$ .

**Definición 20.** Dada una **PTN**  $G = (V, E)$ , con la **EAN** asociada  $\mathcal{N}^0 = (\mathcal{E}^0, \mathcal{A}^0)$ , las cotas  $L_a, U_a, a \in \mathcal{A}^0$ , un periodo de planificación  $T$ , una matriz OD  $OD$ . El Problema integrado de Calendarización de Viajes y Enrutamiento de Pasajeros (**TimPass**) consiste en encontrar un calendario periódico factible  $\pi$  con periodo  $T$  que minimice el tiempo total de viaje en los caminos más cortos con respecto a  $d(\pi)$  para todos los pasajeros, es decir:

$$\min \sum_{(u,v) \in OD} C_{u,v} \cdot \text{len}(SP_{u,v}(d(\pi)), d(\pi))$$

tal que  $\pi$  es un calendario periódico factible con periodo  $T$ .

A continuación, se detalla el modelo **TimPass** presentado en [8]:

$$\min \sum_{(u,v) \in OD} C_{u,v} \cdot \sum_{a=(i,j) \in \mathcal{A}^0} p_a^{u,v} (\pi_i - \pi_j + z_a \cdot T) \quad (2.1)$$

s.t.

$$\pi_i - \pi_j + z_a \cdot T \geq L_a \quad a = (i, j) \in \mathcal{A}^0 \quad (2.2)$$

$$\pi_i - \pi_j + z_a \cdot T \leq U_a \quad a = (i, j) \in \mathcal{A}^0 \quad (2.3)$$

$$A \cdot (p_a^{u,v})_{a \in \mathcal{A}} = b^{u,v} \quad a \in \mathcal{A}, (u, v) \in OD \quad (2.4)$$

$$\pi_i \in \{0, \dots, T - 1\} \quad i \in \mathcal{E}^0$$

$$z_a \in \mathbb{Z} \quad a \in \mathcal{A}^0$$

$$P_a^{u,v} \in \{0, 1\} \quad (u, v) \in OD, a \in \mathcal{A}$$

Donde  $A$  es la matriz de incidencia de la **EAN** extendida  $\mathcal{N}$  y  $b^{u,v}$  el vector de demanda que asegura que el flujo inicie en el nodo de origen y termine en el nodo destino.

La función objetivo puede ser linealizada con la sustitución:

$$d_a^{u,v} = p_a^{u,v} \cdot (\pi_j - \pi_i + z_a \cdot T)$$

con variables enteras auxiliares  $d_a^{u,v} \geq 0$ ,  $a = (i, j) \in \mathcal{A}^0$ ,  $(u, v) \in OD$ , con:

$$d_a^{u,v} \geq \pi_j - \pi_i + z_a \cdot T - (1 - p_a^{u,v}) \cdot M \quad a = (i, j) \in \mathcal{A}^0, (u, v) \in OD$$

donde  $M$  es suficientemente grande, por ejemplo  $M \geq \max_{a \in \mathcal{A}^0} U_a$

**Nota:** Si bien el modelo **TimPass** presentado por [8] no considera el caso de líneas con frecuencias mayores que 2, siguiendo la sugerencia presentada en la **Definición 10**, es posible incorporar fácilmente al modelo tomando en cuenta que las líneas no pueden ocupar el mismo espacio físico durante su operación.

## 2.4. Reducción del tamaño del problema

Para ilustrar las dificultades causadas por el tamaño del problema considere como ejemplo la instancia **Pequeña 2** presentada en [4], que consta de 10 estaciones, un conjunto de 7 líneas con sus frecuencias y 79 pares OD, la **PTN** se encuentra en la figura 2.4. Como se observa, la estructura que tiene la red es pequeña para fines reales; sin embargo, el número de variables necesarias para formular el modelo Tim-Pass resulta ser considerablemente alto.

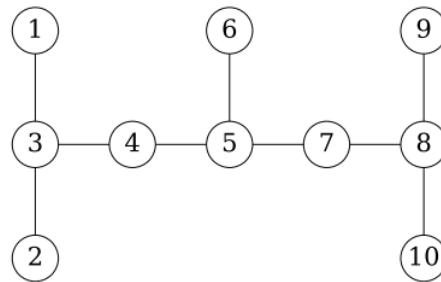


Figura 2.4: **Instancia Pequeña 2** [4]

En la tabla 2.4 se observa el número de variables que se crean para resolver el modelo, el cual es considerable para el análisis pues se



	<b>Pequeña 2</b>
<b>PTN:</b> $ V $	10
<b>PTN:</b> $ E $	9
<b>EAN:</b> $ \mathcal{E} $	548
<b>EAN:</b> $ \mathcal{A} $	9422
<b>Pares OD</b>	79
<b>Vars:</b> $\pi$	528
<b>Vars:</b> $p_a^{u,v}$	74338
<b>Vars:</b> $d_a^{u,v}$	74338
<b>Vars:</b> $z_a$	8892
<b>Variables</b>	205902

Cuadro 2.1:

trata de un ejemplo pequeño. En [4] se muestra que incluso no llega a resolverse hasta la optimalidad con un tiempo límite de ejecución de 7200 segundos. En la citada implementación de la instancia se usó como herramienta la API Python del software comercial de programación entera Gurobi versión 9.5.1 ejecutados en un sistema operativo Linux CentOS 6 de 4 procesadores Xeon de 2.6 GHz, 32 GB de RAM, obteniéndose los siguientes resultados:

	<b># Restr</b>	<b># Vars</b>	<b># Nod</b>	<b>GAP</b>	<b>FunObj</b>	<b>CotInf</b>	<b>Tim</b>
<b>Pequeña 2</b>	76 3770	1 498 098	1	48%	25 301	48 632	7200

Cuadro 2.2:

Se muestra entonces que resolver el **TimPass** de manera óptima ya es difícil incluso para instancias pequeñas, debido al número de variables necesarias para la creación del modelo. En la práctica llegar a subproblemas que se encuentran ligados a un tiempo de cómputo malo, no es bueno y peor aún cuando se tienen instancias cuyas restricciones y variables copan la memoria del computador como por ejemplo la instancia **Grid** que se presentará más adelante.

Como se mencionó en un inicio, el sector público necesita tomar decisiones importantes y encontrarse con problemas como los que se han descrito puede ocasionar cuestionamientos acerca del uso de la tecnología e incluso del de la matemática aplicada, lo que se constituye en una razón adicional para la búsqueda de alternativas que relajen el proble-

ma, además del interés propio en encontrar soluciones al mismo. En el presente capítulo se presentarán dos enfoques de [8] y [3] para reducir el tamaño del problema para que sea computacionalmente más accesible.

### 2.4.1. Heurística LB

Antes de introducir las ideas para reducir el tamaño del problema consideremos la siguiente notación junto con algunos resultados:

**Lema 1.** Sea  $\pi$  un calendario factible. Entonces se tiene que:

$$\text{len}(SP_{u,v}(L), L) \leq \text{len}(SP_{u,v}(d(\pi)), d(\pi)) \quad (2.5)$$

**Definición 21.** Sea  $\pi$  un calendario, sea  $\overline{OD} \subseteq OD$  un subconjunto de pares OD.

- El objetivo de ruteo **Camino más corto** con respecto a  $\overline{OD}$  es:

$$\mathcal{R}_{SP}(\overline{OD}, \pi) = \sum_{(u,v) \in \overline{OD}} C_{u,v} \cdot \text{len}(SP_{u,v}(d(\pi)), d(\pi))$$

- El objetivo de ruteo **cota inferior** con respecto a  $\overline{OD}$  es

$$\mathcal{R}_{LB}(\overline{OD}, \pi) = \sum_{(u,v) \in \overline{OD}} C_{u,v} \cdot \text{len}(SP_{u,v}(L), d(\pi))$$

Observe que con la **Definición 21** se pueden reformular los modelos **PESP** y **TimPass**. En el siguiente lema se presenta una relación entre la función objetivo de los dos problemas evaluados en el óptimo.

**Lema 2.** Sea  $\pi$  una solución óptima del **PESP** y  $\pi^*$  la solución óptima del **TimPass**, entonces,

$$\mathcal{R}_{SP}(OD, \pi^*) \leq \mathcal{R}_{SP}(OD, \pi) \leq \mathcal{R}_{LB}(OD, \pi)$$

Además, con respecto a la brecha de optimalidad entre el objetivo de ruteo Camino más corto y el objetivo de ruteo cota inferior, se tiene el siguiente

resultado:

**Teorema 1.** Considere una instancia de la calendarización de viajes con una **EAN** extendida  $\mathcal{N} = (\mathcal{E}, \mathcal{A})$ ,  $U_a - L_a \leq T - 1$ ,  $a \in \mathcal{A}$  y  $L_a \geq 1$  para todo  $a \in \mathcal{A}$  si  $L_a \neq U_a$ . Sea  $\pi$  una solución óptima para el **PESP** y  $\pi^*$  un calendario de viajes factible para **TimPass**. Entonces, se satisface la siguiente desigualdad:

$$\frac{\mathcal{R}_{SP}(\pi)}{\mathcal{R}_{SP}(\pi^*)} \leq \frac{\mathcal{R}_{LB}(\pi)}{\mathcal{R}_{SP}(\pi^*)} \leq T$$

Ahora, como se menciona en [3], el principal factor por el que el problema **TimPass** es muy grande para resolverlo de manera óptima, es el número de pares OD que se incluyen en el modelo, pues como se observa en la restricción 2.4, por cada par OD se crea una nueva variable.

Surge entonces, lo que se propone en [3]: reducir el número de pares OD a ser ruteados en el modelo **TimPass**; por ejemplo, considerando que el conjunto  $OD_{route} \subseteq OD$  contenga 10% de los elementos de  $OD$ . Más adelante se presentarán algunas formas de escoger dichos pares, por ahora se presenta la primera heurística:

---

**Algorithm 1** Heurística LB

---

**Require:** EAN  $\mathcal{N}^0 = (\mathcal{E}^0, \mathcal{A}^0)$ , cotas  $L_a, U_a$  con  $a \in \mathcal{A}^0$ ,  $OD_{route} \subseteq OD$

**Ensure:** Calendario factible  $\tilde{\pi}$

1. Calcular la solución óptima del Problema (**TimPass**( $OD_{route}$ )) para  $OD_{route}$  y  $\mathcal{R}_{SP}(OD_{route}, \tilde{\pi})$
- 

Si bien la Heurística LB [1] ayuda a eliminar variables asociadas a pares  $OD$  que no son interés, pueden seguir existiendo dificultades en encontrar de manera eficiente resultados óptimos o que tengan una brecha de optimalidad aceptable. Más allá de ello, las soluciones que se obtiene de la **Heurística LB** no aportan calendarios factibles al modelo **TimPass**, esto se debe que al momento de seleccionar  $\overline{OD} \subseteq OD$  se ignora por completo a los pasajeros y podría resultar que tengan tiempos de viaje demasiados largos. En la siguiente sección se presentará un enfoque para compensar los pares  $OD$  que se están ignorando.

## 2.4.2. Heurística UB

Como se mencionó en la sección anterior, la **Heurística LB** ignora por completo un conjunto de pares  $OD$ , en esta parte se propone una manera más realista para evaluar el **TimPass**, para ello se define lo siguiente:

**Definición 22.** Sea  $\overline{OD} \subseteq OD$  cualquier subconjunto de pares  $OD$ . Un límite inferior en el tiempo de viaje mínimo que necesitan los  $OD$  pares en  $\overline{OD}$  en cualquier horario factible, puede definirse por

$$\tilde{L}(\overline{OD}) = \sum_{(u,v) \in \overline{OD}} C_{u,v} \cdot \text{len}(SP_{u,v}(L), L)$$

es decir:

**Lema 3.** Sea  $\overline{OD} \subseteq OD$  cualquier subconjunto de pares  $OD$ . Entonces,

$$\tilde{L}(\overline{OD}) \leq \mathcal{R}_{SP}(\overline{OD}, \pi)$$

para cualquier calendario factible.

**Demostración:** Se obtiene directamente del **Lema 1**.  $\square$

Ahora, para usar  $\tilde{L}(\overline{OD})$  con los pares  $OD$  ( $\overline{OD} = OD_{fix}$ ) y para fortalecer la cota inferior dada por la Heurística LB. Se define:

$$h(OD_{route}, \pi) = \mathcal{R}_{SP}(OD_{route}, \pi) + \tilde{L}(OD_{fix})$$

Lo cual produce una mejor aproximación de  $\mathcal{R}_{SP}(\pi)$

**Lema 4.** Sea  $\tilde{\pi}$  una solución óptima de **TimPass**( $OD_{route}$ ), y sea  $\pi^*$  un calendario factible para el **TimPass**. Entonces, se tiene que:

$$\begin{aligned} \mathcal{R}_{SP}(OD_{route}, \tilde{\pi}) &\leq h(OD_{route}, \tilde{\pi}) \leq \mathcal{R}_{SP}(OD, \pi^*) \\ &\leq \mathcal{R}_{SP}(OD, \tilde{\pi}) \end{aligned}$$

Note que aunque se ha agregado el límite inferior de las mejores rutas en la función objetivo  $h$ , todavía se ignora a los pasajeros que no están

en  $OD_{route}$  durante la optimización y esto puede generar un valor objetivo  $\mathcal{R}_{SP}(OD, \tilde{\pi})$  que puede ser mucho peor que  $\mathcal{R}_{SP}(OD, \pi)$  de una solución óptima  $\pi$  del problema **PESP** como se puede ver en la sección experimental de [3].

Por tanto, la heurística es inútil para generar una buena solución, sin embargo puede usarse como buen límite inferior en **TimPass**

Una mejor idea es incluir los pares OD de  $OD_{fix}$ , enrutándolos de antemano en los caminos más cortos con respecto a los límites inferiores de las actividades, y añadir los tiempos de viaje y los pesos a la función objetivo. En [3] se presenta el modelo:

$(TimPass\&Fix(OD_{route}))$

$$\begin{aligned} \min \quad & \sum_{a=(i,j) \in \mathcal{A}^0} (w_a + \sum_{(u,v) \in OD_{route}} C_{u,v} \cdot p_a^{u,v}) \cdot (\pi_j - \pi_i + z_a \cdot T) \\ & \pi_j - \pi_i + z_a \cdot T \geq L_a \quad a = (i, j) \in \mathcal{A}^0 \\ & \pi_j - \pi_i + z_a \cdot T \leq U_a \quad a = (i, j) \in \mathcal{A}^0 \\ & A \cdot (p_a^{u,v})_{a \in \mathcal{A}} = b^{u,v} \quad (u, v) \in OD_{route} \\ & \pi_i \in \{0, \dots, T-1\} \quad i \in \mathcal{E}^0 \\ & z_a \in \mathbb{Z} \quad a \in \mathcal{A}^0 \\ & p_a^{u,v} \in \{0, 1\} \quad (u, v) \in OD_{route}, a \in \mathcal{A} \end{aligned}$$

Resolviendo  $(TimPass\&Fix(OD_{route}))$  se conduce a la segunda heurística que obtiene buenos límites superiores:

---

**Algorithm 2** Heurística UB

---

**Require:** EAN  $\mathcal{N}^0 = (\mathcal{E}^0, \mathcal{A}^0)$ , cotas  $L_a, U_a$  con  $a \in \mathcal{A}^0$ ,  $OD_{route} \subseteq OD$

**Ensure:** Calendario factible  $\pi$ , evaluación de  $f(OD_{route}, \pi)$

1. Calcular la solución óptima del Problema **(TimPass&Fix(OD<sub>route</sub>))** para  $OD_{route}$  y  $f(OD_{route}, \pi) = \mathcal{R}_{SP}(OD_{route}, \pi) + \mathcal{R}_{LB}(OD \setminus OD_{route}, \pi)$
- 

Note que cuando  $OD_{route} = OD$  se tiene el problema **TimPass** donde todos los pasajeros son ruteados en la optimización, y cuando  $OD_{fix} = OD$  se tiene **PESP**. Si bien esta heurística nos ayuda a tener mejores resulta-

dos que el **TimPass** y el **PESP**, computacionalmente sigue teniendo un costo computacional alto, es por ello que en la sección experimental no será abordado, pero aún se tiene una buena alternativa de reducción de tamaño como lo es la siguiente sección.

### 2.4.3. Algoritmo de preprocesamiento

Cuando se integran las fases de ruteo de pasajeros y calendarización, en el modelo **TimPass** se crean variables de flujo para todos los pasajeros y todas las actividades. Cuando los pares OD usan los caminos más cortos, algunas actividades pueden ser mejoradas de antemano. Por ejemplo, supongamos un par OD viajando desde Quito hasta Guayaquil, éste muy probablemente en el camino más corto no pasará por Loja, sin importar el calendario que se escoja.

Para cada actividad  $a \in \mathcal{A}$ , se sabe que no es parte del camino más corto de  $(u, source)$  hasta  $(v, target)$  para algún calendario si el mejor caso de camino más corto desde  $(u, source)$  hasta  $(v, target)$  que pasa por  $a$  es más grande que el peor de los casos de camino más corto de  $(u, source)$  a  $(v, target)$  sin tener que usar el arco  $a$ . Aquí, el mejor y el peor de los casos depende del calendario, es decir el mejor de los casos de camino más corto es el camino  $(u, source) - (u, target)$  más corto para el calendario que es el mejor para este par OD, mientras que el camino más corto en el peor de los casos es el camino  $(u, source) - (v, target)$  más corto en el peor calendario posible.

Como construir un calendario factible es NP-completo (Serafini and Ukovich 1989), la construcción del mejor o peor calendario también lo es. En lugar de ello, se usará los límites superior e inferior de las actividades como límites de la longitud de las rutas más cortas en el mejor y peor de los casos.

**Teorema 2.** Sea  $(u, v) \in OD$  un par OD, y sea  $a = (i, j) \in \mathcal{A}$  una actividad tal que:

$$len(SP_{u,v}(U), U) < len(SP_{u,i}(L), L) + L_a + len(SP_{j,v}(L), L)$$

Entonces, para cada calendario  $\pi$ , ningún camino más corto  $SP_{u,v}(d(\pi))$  con respecto a  $\pi$  contiene la actividad  $a$ .

Con el teorema anterior en [3] se propone el Algoritmo 3 que determina, para cada par OD, una lista de actividades que nunca serán parte de un camino más corto para este par OD.

---

**Algorithm 3** Algoritmo de procesamiento para el modelo integrado **Tim-Pass** para el par OD  $(u, v)$

---

**Require:** EAN extendida  $\mathcal{N} = (\mathcal{E}, \mathcal{A})$ , asociada a la PTN  $(V, E)$ , cotas  $L_a, U_a$  con  $a \in \mathcal{A}$ , estación de inicio  $u \in V$  y estación de llegada  $v \in V$

**Ensure:** Lista de actividades  $\bar{\mathcal{A}}$  que pueden ser descartadas, es decir que no están contenidas en el camino más corto desde  $(u, source)$  hasta  $(v, target)$  para cualquier calendario factible.

1. Inicializar  $\bar{\mathcal{A}} = \emptyset$
  2.  $\triangleright$  Calcular el camino más corto en el peor de los casos.
  3. Calcular  $\beta := len(SP_{u,v}(U), U)$
  4.  $\triangleright$  Calcular el camino más corto en el mejor de los casos.
  5. **for** evento  $i \in \mathcal{E}$  **do**
  6.     Calcular  $\gamma_i := len(SP_{u,i}(L), L)$
  7.     Calcular  $\delta_i := len(SP_{i,v}(L), L)$
  8. **end for**
  9. **for** actividad  $a = (i, j) \in \mathcal{A}$  **do**
  10.     **if**  $\gamma_i + L_a + \delta_i > \beta$  **then**
  11.          $\bar{\mathcal{A}} = \bar{\mathcal{A}} \cup \{a\}$
  12.     **end if**
  13. **end for**
- 

Como se menciona en [3], en algunos enfoques de calendarización de viajes se consideran las actividades de viaje y espera fijos, en esta publicación se muestra el ejemplo de una red de trenes en Alemania, que dado un número significativo de duraciones de actividades fija y únicamente actividades de transferencia variable (restringido a intervalos) provoca que disminuya el tamaño del problema y aumente el efecto algorítmico de preprocesamiento como se puede ver en 2.5.

Es decir, el algoritmo de preprocesamiento es significativamente más eficiente para duraciones de espera y viaje fijas, pero este tópico no se aborda en el presente trabajo.

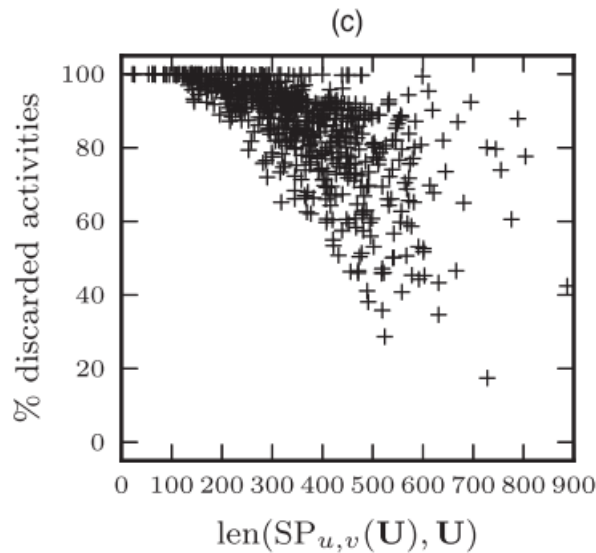


Figura 2.5: Algoritmo de preprocesamiento instancia long-distance [3]

#### 2.4.4. Selección de pares OD

En las secciones anteriores se ha determinado que las heurísticas **LB** y **UB** necesitan un conjunto  $OD_{route} \subseteq OD$ ; la pregunta natural que surge es cómo seleccionar el  $OD_{route}$ . En esta sección se mostrarán tres métodos que se usarán en la sección experimental para la selección del conjunto de interés. Estos son:

- Escoger los **k** pares OD con mayor demanda
- Escoger los **k** pares OD con menor demanda.
- Seleccionar **k** pares OD aleatorios.



# Capítulo 3

---

## Resultados computacionales

---

### 3.1. Descripción de las instancias

Las instancias que se usaron para la experimentación fueron 3: 2 de ellas provienen del Software [5] cuyos datos se los encuentra en [6], para realizar un enfoque ilustrativo se utilizó una instancia pequeña de [4]. El conjunto de datos necesarios para la implementación de la heurística **LB** son las mismas que se necesitan para calcular **TimPass**, es decir: periodo de planificación **T**, conjunto de paradas de la **PTN**, conjunto de líneas y sus respectivas frecuencias que para este trabajo fueron obtenidas de [1], tiempo mínimo y máximo de espera, transbordo y viaje, pares OD con sus respectiva demanda. Los recursos utilizados para la obtención de los resultados presentados en este capítulo son entregados como parte de la componente práctica de la asignatura.

#### 3.1.1. Instancia Pequeña 1

Esta instancia fue tomada de [6], donde la **PTN** consta de  $|V| = 8$  y  $|E| = 8$ , con 6 líneas y las respectivas frecuencias y 46 pares OD. Gráficamente en 3.1

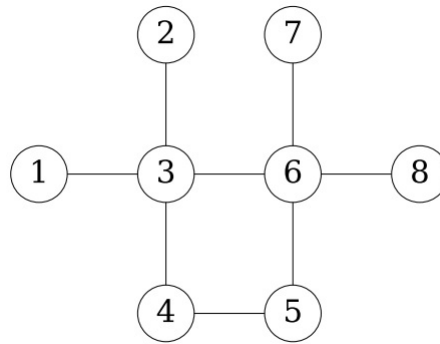


Figura 3.1: Gráfico de la **PTN** de la instancia **Pequeña 1**

### 3.1.2. Instancia Pequeña 2

Esta instancia fue tomada de [4], la **PTN** consta de  $|V| = 10$  y  $|E| = 9$ , con 6 líneas y las respectivas frecuencias y 46 pares OD. Gráficamente en 2.4

### 3.1.3. Instancia Grid 5x5

Finalmente, de [6] se extrae la instancia cuya **PTN** consta de  $|V| = 25$  y  $|E| = 40$ , con 11 líneas y las respectivas frecuencias y 567 pares OD. Gráficamente en 3.2

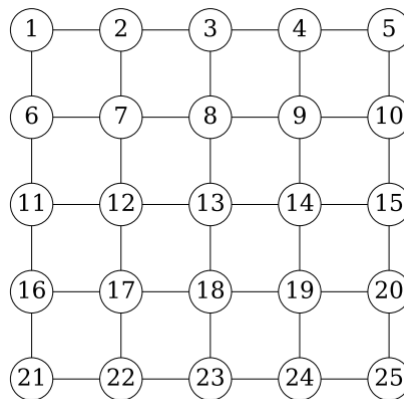


Figura 3.2: Gráfico de la **PTN** de la instancia **Pequeña 1**

## 3.2. Resultados y análisis:

La implementación de las heurísticas fueron implementadas con la herramienta API Python del software comercial de programación entera Gurobi versión 9.5.2. en una computadora con sistema operativo Windows 10 de 64 bits, procesador basado en x64, procesador Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz 3.19 GHz, con RAM instalada de 16,0 GB (15,8 GB usable). Para todas las instancias se definieron el parámetro **TimeLimit** con un tiempo de 1200 segundos (20 minutos) para comparar resultados de la heurística **LB** y el algoritmo de preprocesamiento seleccionando de 3 maneras distintas conjuntos OD, con  $|OD| = k$  y  $k = 1, 10, 50$  que en total fueron aproximadamente 20 horas de cómputo.

### 3.2.1. Resultado Ilustrativo: Instancia Juguete

Esta instancia fue escogida con la intención de ilustrar los efectos de las heurísticas sobre las instancias y que éstas sean comparables con el óptimo. Suponga que el criterio para reducir el problema es quitando del problema los orígenes-destino con menos demanda, en este caso hablamos de la selección de los **k** pares OD más grandes, 3.3 fueron los resultados que incluye los resultados del modelo **TimPass**:

Indx	Heur	k	#Constrs	#Vars	NodeCou	MIPGap	ObjBound	ObjVal
75	Preprocess	1	144	156.00	1	899	899	0.082
92	LB	1	272	336.00	1	589	589	0.172
74	Preprocess	10	1,160	1,490.00	4,530	2105	2,105	2.334
91	LB	10	1,424	2,352.00	1	2516	2,516	0.191
73	Preprocess	Todos	2,820	3,850.00	325,929	4888	4,888	241.736
93	TimPass	Todos	3,344	5,712.00	1	4067	4,067	0.275
90	LB	Todos	3,344	5,712.00	1	4067	4,067	0.605

Figura 3.3: Resultados: Instancia Juguete

- Note que los experimentos asociados al algoritmo de preprocesamiento ayudan a descartar considerablemente las actividades del modelo **TimPass** y comparado a la heurística **LB** podemos concluir que el tamaño del problema con el algoritmo de preprocesamiento es más pequeño que el de la heurística **LB**.

- El peor tiempo de cómputo de una solución óptima está en el algoritmo de preprocesamiento cuando se selecciona quitar del modelo actividades, sin embargo es la que más se aproxima a la función objetivo con una diferencia de 821.
- Se descarta **LB** con  $k = 25$  pues recuerde que cuando  $OD_{route}$  es tan grande como  $OD$  se está calculando **TimPass**.

### 3.2.2. Resultados Instancia Pequeña 2

#### Resultados para las instancias que alcanzan el óptimo:

Como se puede observar en 3.4, la mayoría de instancias que logran llegar al óptimo en menos de 1200 segundos son las que admiten un número de pares OD igual a 1, además, note que la selección de 10 pares OD aleatorios junto con el preprocesamiento logran llegar al óptimo en menor tiempo del fijado, que no es el caso de los experimentos con alguna selección  $k = 10$  de pares similar que se tienen más adelante.

Por otra parte, las dos instancias que llegan a la solución óptima consideran el mismo par OD, tienen el mismo valor objetivo pero, en porcentaje el preprocesamiento ayuda a encontrar más rápido la solución.

En la selección del par OD más pequeño se puede observar que si bien el algoritmo de preprocesamiento ayuda a eliminar variables y restricciones del modelo, en este caso la elección del par más pequeño para la heurística **LB** resulta mejor.

Indx	Heur	M.OD	k	#Constrs	#Vars	NodeCoun	MIPGap	ObjBound	ObjVal	Tim
45	Preprocess	Grande	1	1,153	1,489	1	0.00%	1050	1,050	0.226
34	LB	Grande	1	22,362	23,226	1,589	0.00%	1050	1,050	42.609
48	Preprocess	Pequeño	1	14,174	14,263	3,820	0.00%	21	21	230.469
35	LB	Pequeño	1	22,362	23,226	3,533	0.00%	14	14	75.547
51	Preprocess	Rand	1	14,174	14,244	1,296	0.00%	930	930	190.092
36	LB	Rand	1	22,362	23,226	1,386	0.00%	690	690	160.144
50	Preprocess	Rand	10	53,768	89,174	933	0.00%	4320	4,320	1,002.853

Figura 3.4: Resultados: Instancias que alcanzan el óptimo

**Resultados que no alcanzan el óptimo:** En esta sección la experimentación fue detenida por el criterio que se dio al programa, de manera que se va a comparar, el resultado con mejor brecha de optimalidad con

el experimento que tenga las mismas condiciones en la elección de los pares y difiera en el método de cálculo usado, que en este caso son los experimentos número **39** y **46** que como se puede notar el preprocesamiento ayudó primero a reducir las variables del problema y a reducir la brecha de optimalidad.

Ahora, para los experimentos (**43** y **37**) con la elección de  $k = 25$  más grandes tienen un comportamiento muy parecido en el GAP incluso cuando las actividades que se descartaron del preprocesamiento lograron reducir el número de variables y restricciones.

Indx	Heur	M.OD	k	#Constrs	#Vars	NodeCoun	MIPGap	ObjBound	ObjVal	Tim
44	Preprocess	Grande	10	58,696	98,840	1,285	7.52%	9840	10,640	1,210.368
38	LB	Grande	10	92,220	162,582	4,233	17.43%	6160	7,460	1,200.420
47	Preprocess	Pequeño	10	58,726	99,120	6,408	27.31%	1115	1,534	1,200.240
40	LB	Pequeño	10	92,220	162,582	3,906	18.76%	840	1,034	1,209.506
42	LB	Rand	10	84,458	147,098	3,942	28.64%	2940	4,120	1,201.466
43	Preprocess	Grande	25	133,136	241,760	37	49.60%	28390	56,330	1,200.783
37	LB	Grande	25	208,650	394,842	544	48.62%	17480	34,020	1,201.460
46	Preprocess	Pequeño	25	133,036	240,610	164	1.79%	4566	4,649	1,200.518
39	LB	Pequeño	25	208,650	394,842	245	44.07%	2519	4,504	1,200.994
49	Preprocess	Rand	25	93,386	164,656	1,263	2.93%	6960	7,170	1,203.757
41	LB	Rand	25	146,554	270,970	1,614	27.84%	4820	6,680	1,205.567
54	Preprocess	Grande	50	257,036	481,260.00	1	79.49%	38020	185,380	1,200.125
57	LB	Grande	50	402,700	781,942.00	1	80.90%	24520	128,380	1,200.195
53	Preprocess	Pequeño	50	257,036	481,260.00	1	66.27%	16382	48,569	1,200.220
56	LB	Pequeño	50	402,700	781,942.00	1	77.80%	10091	45,463	1,200.193
52	Preprocess	Rand	50	187,544	346,700.00	8	50.25%	18112	36,405	1,202.150
55	LB	Rand	50	294,032	565,166.00	58	69.70%	11749	38,777	1,200.368

Figura 3.5: Resultados que no alcanzan el óptimo:

### 3.2.3. Resultados Instancia Grid 5x5

#### Resultados que no alcanzan el óptimo:

El resultado que se quiere destacar de 3.6 en esta sección es el número de restricciones y variables que tiene el experimento de evaluar la instancia **Grid 5x5** en el modelo **TimPass** (fila de color rojo) que, como se puede observar, tiene una brecha de optimalidad y función objetivo altos.

En este caso el mejor experimento con GAP bajo es el experimento **30** que es afectado por el algoritmo de preprocesamiento y comparando con los otros dos métodos de elección de  $k = 25$  pares OD, en este caso el

método de escoger pares aleatorios resulta ser beneficioso.

Indx	Heur	M.OD	k	#Constrs	#Vars	NodeCou	MIPGap	ObjBound	ObjVal	Tim
1	-	-	625	1,795,966	3,527,580	1	inf	18227	inf	1,200.212
22	Preprocess	Grande	25	60,433	100,697	13,301	27.98%	7236	10,047	1,200.283
27	Preprocess	Pequeño	10	28,306	44,016	14,129	14.44%	231	270	1,200.146
26	Preprocess	Pequeño	25	64,716	107,051	6,676	24.05%	537	707	1,200.396
30	Preprocess	Rand	25	55,083	90,340	7,131	13.78%	1170	1,357	1,200.202

Figura 3.6: Resultados que no alcanzan el óptimo:

**Resultados para las instancias que alcanzan el óptimo:** Finalmente, notemos que en los experimentos **16**, **15** y **14**, se encuentran calculados bajo la heurística **LB** sin embargo la elección de los  $k = 25$  pares más grandes resulta provechoso para la selección de esa cantidad de pares.

Indx	Heur	M.OD	k	#Constrs	#Vars	NodeCou	ObjBound	ObjVal	Tim
24	Preprocess	Grande	1	720	1,052	1	1230	1,230	0.140
23	Preprocess	Grande	10	4,822	7,104	1,348	2856	2,856	0.765
4	LB	Grande	1	8,538	9,324	1,308	1968	1,968	9.350
9	LB	Grande	10	36,960	65,268	1	5946	5,946	35.887
14	LB	Grande	25	84,330	158,508	1	10242	10,242	63.013
28	Preprocess	Pequeño	1	59	460	0	6	6	0.028
5	LB	Pequeño	1	8,538	9,324	1	12	12	2.544
10	LB	Pequeño	10	36,960	65,268	1,279	303	303	178.397
15	LB	Pequeño	25	84,330	158,508	1,300	768	768	375.697
8	LB	Rand	1	8,538	9,324	1	108	108	6.032
32	Preprocess	Rand	1	4,350	4,468	289	60	60	8.540
13	LB	Rand	10	30,644	52,836	898	606	606	67.871
31	Preprocess	Rand	10	28,306	44,046	1,920	1572	1,572	188.164
16	LB	Rand	25	81,172	152,292	2,959	2505	2,505	546.087

Figura 3.7: Resultados: Instancias que alcanzan el óptimo

# Capítulo 4

---

## Conclusiones y Recomendaciones

---

- Se comprueba la necesidad de explorar métodos heurísticos para el problema integrado para Calendarización de viajes y Ruteo de Pasajeros, toda vez que resolver el modelo entero asociado es computacionalmente costoso, incluso para instancias pequeñas.
- Como se pudo ver en la sección de resultados computacionales, tanto la heurística **LB** como el algoritmo de preprocesamiento aportan para la reducción del número de variables del problema.
- El tiempo de cómputo continúa siendo una limitante del problema integrado bajo este enfoque, pues si bien hay combinaciones de experimentos de la instancia **Grid 5x5** que llegan al óptimo, no son aplicables a la vida real debido a que si se pone en una balanza los pasajeros en los que se centra la atención para resolver el problema, ignorando a las otras 600 necesidades (como en 3.6), no resulta ser bueno para que un sistema de transporte se considere que ofrece buenos servicios.
- Si bien 1 propone un modelo para no ignorar de cierta manera el conjunto de interés  $OD_{fix}$ , computacionalmente también se tiene un costo alto sin utilizar **Cycle-Base Formulation** presentada en [3].
- Para la implementación del Algoritmo de preprocesamiento se recomienda explorar en el paquete *NetworkX* cuyas funciones son de

gran ayuda para el cálculo de caminos más cortos y longitud de caminos más cortos.

- Para fines prácticos es recomendable considerar alternativas de prioridad con respecto al origen y destino que puede tomar los usuarios, seleccionando un subconjunto de pares OD los cuales tengan mayor demanda, considerando que los pasajeros del conjunto ignorado pueden tener tiempos demasiado largos dentro del sistema de transporte.
- Para futuros trabajos que tengan un enfoque parecido, incluso si se relaciona a la optimización de modelos de programación lineal, se recomienda explorar más en los parámetros del Solver Gurobi como por ejemplo los parámetros **MipFocus** y **Heuristics**.
- Para la implementación del modelo **TimPass** se recomienda considerar a las frecuencias mayor a 1 como copias de la misma, tomando en cuenta el espacio físico que tendrá la red de transporte, o si en algo ayuda, pensar que los buses que operan las líneas no podrán compartir las actividades de llegada y que la diferencia de tiempo entre cada evento tiene que ser a lo más **T**.
- Para obtener las instancias de [5] necesarias para el problema integrado bajo el enfoque que presenta [8] se recomienda depurar información que se encuentran en los datasets, para que la programación pueda ser más sencilla.



---

## Referencias bibliográficas

---

- [1] Darlyn Ludeña. *Programación Lineal Entera para la Planificación de líneas en la optimización de sistemas de Transportación Pública*. 2022.
- [2] R. Giesen J.C Muñoz O.J Ibarra-Rojas, F. Delgado. *Planning, operation and control of bus transport systems: A literature review*. 2015.
- [3] Anita Schöbel Philine Schiewe. *Periodic Timetabling with Integrated Routing: Toward Applicable Approaches*. Articles in Advance, 2020.
- [4] Liseth Recalde. *Modelos de Programación Lineal Entera para la Calendarización de Viajes y Enrutamiento de Pasajeros*. 2022.
- [5] A Schiewe. *Openlintim*, <https://gitlab.rlp.net/lintim/OpenLinTim>. 2022.
- [6] A Schiewe, S Albert, V Grafe, Schiewe, A Shöbel, and F Spühler.
- [7] Marie E. Schmidt. *Integrated Routing Decisions in Public Transportation Problems*. Springer, 2010.
- [8] Philine Shiewe. *Integrated Optimization in Public Transport Planning*. Springer Cham, 2020.
- [9] Victor R. Vidal V. de Werra D. Silver, E. A. *A tutorial on heuristic methods*. 1980.
- [10] Mauricio Yépez. *Modelos Integrados de Optimización de Transporte Público: Problema de asignación de vehículos integrado la Calendarización de Viajes*. 2022.