Optimización de la Programación de despachos de Autobuses: Aproximación de demanda dinámica con un enfoque hacia el usuario

Maria Paula Estupiñan[a], Santiago Gonzalez R.[b]

[a]*Departamento de Ingeniería Industrial , Universidad de los Andes, Bogotá, Colombia.*
[b]*Departamento de Ingeniería civil y ambiental, Universidad de los Andes, Bogotá, Colombia*

Abstract

La planeación táctica es una etapa esencial para la eficiencia de los sistemas de transporte publico. En esta se buscan optimizar tareas que definan la operación y establezcan servicios con minimización de costos tanto para el sistema como para los usuarios.En este articulo abordamos la programación de despacho de autobuses con el objetivo de minimizar los costos en tiempos de espera de los usuarios en un sistema zonal de transporte publico. Partimos de una visión de flujo en redes donde formulamos un modelo de programación lineal mixta con variables binarias de despacho y de flujo de pasajeros en el servicio y en espera. El modelo es implementado computacionalmente donde se encuentran soluciones exactas con solvers comerciales, a su vez, se presenta un algoritmo GRASP para encontrar soluciones aproximadas y reducir el esfuerzo computacional. Finalmente, a partir de información del sistema estratégico de transporte publico Movisinu de Montería, Colombia, es evaluado el modelo.

Keywords: Transporte Publico, Programación de vehículos, operacciones, Optimización , Tiempos de espera, Programación lineal mixta

1. Introducción

Los sistemas de transporte publico cumplen una función vital en el desarrollo económico y en la competitividad regional de las ciudades. Un diseño integral de estos puede abatir múltiples externalidades del transporte congestión, contaminación, siniestralidad vial, entre otros y reducir considerablemente los costos de viaje de los usuarios. Sin embargo, en el entorno dinámico del transporte surgen desafíos complejos relacionados al diseño y operación de la red de servicio, donde de no ser tratados, derivan en ineficiencias para el operador de transporte y afectaciones hacia el usuario.

En América latina, a diferencia del norte global, el proceso de expansión urbana derivado de la rápida industrialización no ha sido acompañado por un progreso económico, social e institucional que certifique bienestar y desarrollo integro a sus ciudadanos [18]. Estos fenómenos de hiperurbanización, han ejercido presión sobre los sistemas de transporte para que otorguen accesibilidad a oportunidades a partir de un diseño de un sistema con altos niveles de servicio y asequibilidad para los potenciales usuarios [16]. Sin embargo, acciones para la mejora de los sistemas, como la integración tecnológica en la operatividad, resultan ausentes en algunos casos de la región. En este articulo, con el objetivo de evaluar la aplicabilidad del modelo desarrollado, sera estudiado el caso de Movisinu , la empresa operadora de transporte publico de Montería, Colombia

Email addresses: mp.estupinan@uniandes.edu.co (Maria Paula Estupiñan), s.gonzalezr1@uniandes.edu.co (Santiago Gonzalez R.)

1.1. Programación de horarios y vehículos en la red.

La eficiencia de los sistemas de transporte público está fundamentalmente determinada por las características del diseño de la red y su plan operativo. A partir de esto, cada operadora de transporte planifica y gestiona su red utilizando distintos sistemas y modelos. Los sistemas de transporte extensos y complejos, como los Sistemas Integrados de Transporte Público (SITP) en Colombia, suelen automatizar estos procesos para adaptarse al comportamiento dinámico de la demanda. En contraste, los sistemas reducidos, como los Sistemas Estratégicos de Transporte Público (SETP), aunque han implementado la automatización en ciertos procesos, aún dependen en parte de configuraciones manuales y arbitrarias, basadas en criterios empíricos basados en la experiencia operativa. En este marco de trabajo se tienen problemas como: el diseño de la red de servicio (TND,por sus siglas en inglés), ajuste de frecuencia (FS,por sus siglas en inglés) , programación horaria de la red (TNT; por sus siglas en inglés), programación de conductor (DSP;por sus siglas en inglés) y la programación de vehículos (VSP),por sus siglas en inglés) [16]. El problema que sera abarcado en este articulo esta asociado al problema de configuración de frecuencias (FSP, por sus siglas en ingles) el cual busca establecer frecuencias óptimas para el despacho de buses. A mayor detalle, el problema de programación horaria (Frecuencia) de la red busca definir horarios de partida y llegada en los distintos nodos de la red, en este caso, satisfaciendo una frecuencia condicionada por los patrones de demanda y la capacidad del sistema [16].

Los problemas de este tipo cuentan con múltiples actores y propiedades interdependientes que hacen que una aproximación integral sea ideal, sin embargo, la complejidad de cada uno de los subproblemas conlleva dificultades algorítmicas y computacionales [16]. Sumado a esto, la fluctuación de la demanda y la incertidumbre ante eventualidades que inciden en esta, también representan un reto. El ecosistema del transporte publico entonces esta compuesto por agentes, donde los operadores buscan reducir costos operacionales, y los usuarios buscan reducir su costo generalizado de viaje. Una configuración con mas buses en operación y una frecuencia mayor podría incrementar considerablemente el nivel de servicio, reducir tiempos de espera y mejorar la satisfacción del usuario, pero a su vez, también ejerce una gran carga sobre los costos operativos del operador del sistema [17] . A partir de esto, surge el reto de reducir los costos operativos del sistema manteniendo un rango de nivel de servicio factible desde la perspectiva de los usuarios.

Nuestro estudio propone abarcar este conflicto bajo la reducción de tiempos de espera en las estaciones buscando la maximización de satisfacción del usuario y estableciendo intervalos óptimos. Esto a partir de la determinación de tasas de llegada al sistema , reconociendo el comportamiento estocástico de la demanda. El aporte de este estudio recae en la formulación de un problema que integra la programación horaria y asignación de vehículos reconociendo los comportamientos fluctuantes de la demanda en distintas franjas horarias con restricciones de saturación por vehículo.

De esta manera en la sección 2 se abarca una revisión de literatura extensa sobre los diferentes acercamientos a los problemas de tipo TNT y VSP. La sección 3 plantea el problema, las variables y parámetros y la formulación matemática . La sección 4 presenta la metodologia planteada para solucionar el modelo y la configuración del caso de estudio. La sección 5 presenta los resultados del experimento y análisis con el caso de estudio de Movisinu. Posteriormente, la sección 6 presenta una discusión acerca del ejercicio realizado. Finalmente, la sección 7 presenta conclusiones y recomendaciones futuras para el trabajo de los problemas de redes de flujo.

2. Revisión de literatura

El Problema de Programación de Horarios y Asignación de Vehículos en Redes de Transporte (TNTSP, por sus siglas en ingles) presenta un desafió integral en la optimización de sistemas de transporte publico, buscando equilibrar la eficiencia operativa con la satisfacción del usuario. Diversos estudios han abordado este problema desde ángulos complementarios, utilizando enfoques metodológicos innovadores para mejorar la gestión de horarios y vehículos.

Frente a la integración del nivel de servicio percibido por el usuario y la operación del sistema, [4], [2] y [11] presentan metodologías que buscan optimizar simultáneamente la satisfacción del usuario y los costos operativos . El trabajo en [4] utiliza el algoritmo Iterated Local Search (ILS) para ajustar los horarios de salida de los vehículos, optimizando las transferencias entre rutas, garantizando la uniformidad de los intervalos entre salidas y reduciendo los tiempos muertos. En un enfoque paralelo, [2] aplican un modelamiento matemático que presenta similitudes con el de [4], sin embargo, para encontrar la solución óptima transforman el problema original aplicando relajación lagrangiana y programación lineal. Finalmente, los autores en [11] desarrollan un método para construir horarios que combinan el concepto de intervalos y cargas uniformes, adaptados a vehículos de diferentes tamaños. El articulo propone una metodología para equilibrar la demanda fluctuante y minimizar los costos operativos ajustando horarios según la demanda acumulativa.

[10] proponen un modelo de programación bi-nivel para la operación regional de autobuses. En el nivel superior, minimizan tanto el numero total de vehículos necesarios como el tiempo de los viajes de retorno, mientras que en el nivel inferior optimizan la programación de horarios para reducir al mínimo el tiempo total de transferencia de pasajeros en las paradas de conexión. El articulo presenta el algoritmo Bi-level Nesting Tabu Search (BNTS) para optimizar la programación de autobuses en dos niveles y emplea el método 2-opt para explorar nuevas soluciones al alterar pares en las series de viajes y depósitos.

Los artículos de [6] y [23] abordan de manera segregada el modelo de programación bi-nivel propuesto por [10]. El trabajo en [6] plantea un modelo donde se minimiza el tiempo total de transferencia de los pasajeros, considerando el tiempo total de espera y el tiempo total de viaje. Utiliza un algoritmo de estructura de árbol para generar rutas, donde se enumeran según las técnicas de partición de conjuntos y las asigna a los vehículos, buscando un equilibrio entre los tiempos de viaje y de espera. Por otro lado, [23] abordan un problema de asignación de vehículos de un solo deposito para lineas (rutas) cruzadas (SDVSP, por sus siglas en ingles), Tomando como función objetivo la minimización del numero total de vehículos utilizados. Su metodología se basa en la teoría de clasificación de trabajos fijos donde se transforma el problema en uno de ordenación de trabajos, modelando los vehículos como procesadores y los viajes como trabajos. Asimismo, [20] proponen un modelo de intervalo fijo bajo un algoritmo FIFO (First in first out) para la programación óptima de vehículos con un solo deposito, donde también se toma un acercamiento de procesadores y trabajos.

El tiempo de espera es una variable critica en la programación de horarios y asignación de vehículos debido a su impacto en el costo generalizado de viaje y satisfacción del cliente. Los autores en [5] abordan el problema optimizando los horarios para facilitar las transferencias y evitar la acumulación de autobuses en nodos clave, mediante el calculo de ventanas de tiempo y la aplicación del Teorema de Sincronización .[9] Amplia esta perspectiva considerando los costos de demora para los pasajeros que viajan antes o después de lo deseado, utilizando modelos de linea y circular para la optimización. El modelo de linea abarca versiones con costos homogéneos y heterogéneos según

las necesidades de los pasajeros, mientras que el modelo circular ofrece flexibilidad con una programación que se repite cada 24 horas, permitiendo a los pasajeros ajustar sus horarios con mayor facilidad.

El diseño de horarios y la asignación de vehículos a menudo se realizan asumiendo el conocimiento previo de las rutas de los usuarios, aunque en realidad estas rutas dependen de horarios aun desconocidos. El trabajo en [3] abarcan este desafió integrando las elecciones de rutas de los usuarios para minimizar el retraso programado, el numero de vehículos necesarios y el costo de las corridas de linea, empleando el método e-constrain t para la optimización multiobjetivo. Este enfoque de variabilidad en la selección de rutas también es explorado por [7], quien propone un método innovador para la planificación del transporte publico. Su metodología se estructura en tres fases: diseño de rutas, conversión a lineas operativas, y optimización de horarios mediante técnicas de emparejamiento y ajustes iterativos para mejorar la eficiencia y la satisfacción del usuario.

El problema de programación de vehículos varia según la cantidad de depósitos y las características de la flota modelada. En la actualidad, se han abarcado múltiples problemas que tratan la implementación de flotas eléctricas en los sistemas de transporte. El articulo de [22] se enfoca en la programación horaria de bloques operativos de buses en un solo patio de vehículos eléctricos (SDEVSP, por sus siglas en ingles). Su objetivo es minimizar costos a través de la minimización de los tiempos de no beneficio, reducir la flota y evaluar la viabilidad de la transición a la energía eléctrica. Para ello, divide el problema en la programación de vehículos eléctricos y el problema de combinación de bloques (BCP, por sus siglas en ingles), utilizando restricciones como la operación al día siguiente considerando los altos tiempos de cargue de la tecnología verde. El BCP se resuelve con algoritmos como Greedy, MILP, DaC y Simulated Annealing, y emplea bloques de viajes consecutivos para reducir la inactividad y el tamaño de la flota. Frente a este mismo paradigma, [19] propone una aproximación multi agente donde buscan minimizar costos de uso y costos de recarga bajo el método de programación lineal entera mixta (MILP, por sus siglas en ingles).

En contraste, los estudios de [13],[8] y [21] abordan la programación de autobuses con múltiples depósitos y flota homogénea.[13] optimizan el equilibrio entre costos operativos y tiempos de espera utilizando técnicas de flujo de red, *Deadheading* para trayectos vacíos y *Shifting Departure Time* para ajustar horarios y reducir la flota. [8] se centran en asignar vehículos a viajes comerciales para minimizar costos totales, utilizando el algoritmo de búsqueda local iterada (ILS, por sus siglas en ingles) para ajustar soluciones iniciales y gestionar costos operativos y de transferencia. [21] optimizan el costo total bajo la aplicación de métodos de generación de columnas y descomposición de inventarios para modelar cada viaje que se recorre en la red, reduciendo el numero de restricciones y variables, y así, mejorando la eficiencia en la resolución del problema maestro.

Para el problema de la programación de autobuses con múltiples depósitos y flota heterogénea (MVT-MDSPV, por sus siglas en ingles), los estudios de [1], [24] y [11] presentan metodologías integradas que consideran la sustitución de vehículos, la cantidad de asientos vacíos , la capacidad de la flota, los tiempos de los pasajeros y los costos operativos. [1] optimizan horarios de vehículos en una red de transporte mediante un modelo de flujo de red en dos capas, organizando conexiones y reflejando el comportamiento fluctuante de la demanda. [24] introducen una variante del Problema de Programación de Vehículos en Múltiples Depósitos (MDVSP, por sus siglas en ingles) con Desplazamiento Controlado de Viajes (MDVSP-CTS,por sus siglas en ingles), que ajusta ligeramente los horarios para reducir costos operativos sin sacrificar el nivel de servicio. Utilizan una matheuristica de dos fases: heurística de generación de columnas para desarrollar horarios iniciales y

un programa entero mixto para ajustar estos horarios. Finalmente, [11] presentan una metodología basada en un modelo de flujo de red de costo mínimo para múltiples tipos de vehículos (MVT-VSP, por sus siglas en ingles). Utilizan horarios óptimos de Pareto y ajustan la programación según la flota disponible, buscando minimizar los costos operativos mediante la optimización de la elección y sustitución de vehículos, y reduciendo costos por tiempos de espera y asientos vacíos.

Con el objetivo de analizar enfoques centrados en la satisfacción del usuario dentro del sistema, los estudios [25] y [26] proponen formulaciones orientadas a minimizar el tiempo de espera de los pasajeros. [26] plantean una aproximación donde son ajustados los intervalos de despacho de trenes a partir de los patrones de demanda. Estos son ajustados bajo una aproximación de minimización de tiempo de espera de los usuarios del sistema, para esto, se plantea un modelo no lineal (MINLP, por sus siglas en ingles) donde se minimiza el tiempo y un modelo lineal (MILP, por sus siglas en ingles) con el objetivo de minimizar los pasajeros en espera.[26] utilizan una aproximación "Mini-Max" para definir un limite superior del tiempo de espera y simular las elecciones de los pasajeros en el mercado del transporte. Los modelos son simulados con solvers comerciales como KNITRO/GAMS o CPLEX, sin embargo, son desarrolladas heuristicas basadas en variantes del algoritmo de búsqueda de vecindad para resolver el problema. A su vez, el estudio de [25] reconoce la demanda dinámica, y propone tres formulaciones lineales para abordar el problema. Las formulaciones son implementadas en IBM CPLEX y solucionadas con un algoritmo Branch-and-cut.

Para la resolución del proyecto, fueron identificados siete de los artículos revisados que proporcionan herramientas clave para el planteamiento del modelo matemático y la selección de la metodología a emplear. En primer lugar, definimos nuestra función objetivo, revisando la literatura existente. Con base en esta revisión, consideramos que es relevante enfocarnos en la satisfacción del usuario y la eficiencia del servicio. Los trabajos [12], [13], [25] y [26], ofrecen enfoques valiosos para medir la satisfacción del pasajero y evaluar la eficiencia del servicio. Los estudios consideran el tiempo de espera en las paradas de autobús como un factor clave en la satisfacción del usuario. Sin embargo, mientras que el estudio de [13] se centra únicamente en el tiempo en las paradas principales**,** [26] toma el tiempo de espera general del sistema, cargando los pasajeros en intervalos durante un periodo de análisis. En el presente trabajo consideramos relevante asumir el enfoque de [26] dada la limitación y falta de detalle de la información de la demanda del sistema estudiado.

En cuanto a la eficiencia del servicio, se han propuesto diversas metodologías para optimizar la asignación y sincronización de autobuses. [12] utilizan el método de carga máxima para determinar la cantidad óptima de autobuses necesarios en función de la demanda de cada línea y la capacidad de los vehículos. Este enfoque se centra en ajustar la flota de autobuses según la demanda en diferentes momentos del día. Complementariamente, [11] emplean la función de déficit (DF) para minimizar la cantidad de vehículos. La DF calcula y visualiza la cantidad de vehículos requeridos en cada terminal a lo largo del tiempo, aumentando con cada salida de viaje y disminuyendo con cada llegada. Este enfoque permite identificar el número máximo de vehículos necesarios según la franja horaria y facilita la visualización de cómo se distribuyen las necesidades de vehículos a lo largo del día.

A pesar de que aquí no implementamos la eficiencia del servicio como función objetivo, reconocemos la importancia de definir la frecuencia de los autobuses, ya que esto impactaría directamente en el tiempo de espera de los pasajeros. De los artículos revisados, tres destacan explícitamente el enfoque utilizado para definir el ciclo o frecuencia en el desarrollo de modelos matemáticos. El estudio **en** [11] presenta un método para construir horarios con intervalos uniformes y vehículos de

capacidad uniforme. El trabajo en [1] se enfoca en minimizar la desviación de los intervalos establecidos respecto a un intervalo uniforme deseado, así como la desviación de las cargas de pasajeros observadas respecto a un nivel uniforme deseado en el punto de máxima carga. Finalmente,[15] utiliza un enfoque basado en un horario inicial cíclico para ajustar de manera precisa la oferta a la demanda fluctuante.

En este trabajo optamos por definir una frecuencia de autobuses regular con una carga uniforme según franja horaria, lo que simplificaría la planificación de horarios, mejoraría la coordinación de los autobuses y optimizaría los recursos. De las metodologías propuestas en estos artículos, la única que se enfoca exclusivamente en garantizar la uniformidad de los intervalos es la metodología de [11]. Esta metodología crea una curva acumulativa basada en la demanda de pasajeros y la carga observada a lo largo del tiempo, ajustando los horarios de salida de los autobuses a esta curva para asegurar intervalos y carga uniforme.

3. Formulación del problema

El problema se abordará mediante una simulación detallada de la operación diaria de un sistema de transporte público, enfocándonos en la optimización de frecuencias en las rutas de autobuses. Diferenciaremos las condiciones operativas según franjas horarias—específicamente entre horas pico y horas valle—para capturar las variaciones en la demanda de pasajeros. La función objetivo propuesta busca maximizar la satisfacción de los usuarios mediante la reducción del número de pasajeros en espera y, por ende, de los tiempos totales de espera en el sistema.

Esta aproximación tiene como meta establecer frecuencias óptimas de los servicios que permitan una saturación eficiente de los autobuses, considerando las tasas de llegada acotadas por los diferentes regímenes de demanda. Al ajustar adecuadamente las frecuencias, se busca equilibrar la oferta con la demanda, minimizando los costos operativos dependientes del tiempo y evitando tanto la subutilización como la sobrecarga de los vehículos.

Nuestra metodología se centra en la minimización del costo generalizado de viaje de los pasajeros, lograda a través de la reducción del tiempo total de viaje, incluyendo tiempos de espera y desplazamiento. Simultáneamente, se optimizan los recursos del operador mediante la utilización óptima de la capacidad de los autobuses y la programación eficiente de los servicios. De este modo, la asignación de frecuencias puede ser automatizada y ajustarse dinámicamente en respuesta a la incertidumbre y fluctuaciones en la demanda de los sistemas de transporte. Este enfoque no solo mejora la calidad del servicio para los usuarios, sino que también promueve una gestión más rentable y adaptable para los operadores.

La formulación matemática parte de una formulación temporal del flujo en redes. En esta, se modelan los nodos como el inicio de los intervalos de despacho de la línea estudiada, mientras que los arcos representan los intervalos de tiempo entre despachos. A partir de esto, se definen cuatro conjuntos: el primero es el conjunto de intervalos $[t \in T]$ en el período de análisis $[0, |T|]$. Dado el planteamiento del proyecto, se realiza una división de 10 minutos de intervalos de análisis α. El segundo conjunto está formado por los buses disponibles en la línea, $[i \in B]$. El tercer set corresponde a los servicios j realizados por el bus i durante periodo de análisis $[0, |T|]$. Por ultimo, el cuarto set define el sentido del trayecto del bus f, donde 1 indica el trayecto de ida y 2 el trayecto de vuelta $[f \in F]$.

El tiempo total de análisis por el cual se programara el despacho de buses según la franja de demanda analizada dividido en intervalos α de 10 minutos con el objetivo de analizar el flujo de pasajeros. Con esto obtenemos el valor de m. Durante el periodo T se tomara una tasa λ_t que define el arribo deterministico de los usuarios al sistema. Con el objetivo de estimar la frecuencia del despacho de buses se establecen dos parámetros que definen el intervalo mínimo $h_{\min}$ (estimado como el tiempo mínimo de espera por pasajero) y un intervalo máximo $h_{\max}$ (estimado como el tiempo máximo de un pasajero fiel). C representa la capacidad de un bus, mientras que el tiempo $t_{\min}$ representa el tiempo de maniobra realizado por el bus al final de la ruta y el tiempot_{rut} representa el tiempo de la linea entre la estación inicial y final.

Las variables de decisión del modelo son mixtas, donde la variable binarias $x^t_{i,j,f}$. establecen el despacho de un bus, tomando el valor de 1 cuando el bus i sirviendo el servicio j en el sentido f es despachado en el tiempo t. Por otro lado, la variables $b_{t,f}$ representa el numero de pasajeros que abordan el sistema en el intervalo $[t, t+1]$. De la misma manera, la variable $w_{t,f}$ representa los pasajeros que esperan durante el intervalo $[t, t+1]$. A partir de esto la función objetivo es definida en Eq (1) como la minimización de los pasajeros en espera durante el periodo de análisis T.

SETS	
B	Conjunto de buses disponibles en la línea; i: bus
F	Conjunto de sentidos de la línea; f: sentido
S	Conjunto de servicios que puede cumplir el bus i durante T
T	Conjunto de intervalos durante el periodo de análisis; t: intervalo
PARÁMETROS	
C	Capacidad de buses
$h_{\min}$	Intervalo mínimo
$h_{\max}$	Intervalo máximo
m	Número de intervalos en el periodo de análisis
n	Tamaño de la flota
s	Número de servicios
$t_{\min}$	Tiempo de maniobra al final de la ruta
t_{rut}	Tiempo de ruta desde la primera estación hasta la final
α	Intervalos de análisis; periodos donde se carga el sistema
$\lambda_{\text{t,f}}$	Tasa de llegada de pasajeros en el intervalo t
VARIABLES DE DECISIÓN	
$x^t_{i,j,f}$	Variable binaria que define el despacho del bus i en el sentido f en el tiempo t
$b_{t,f}$	Pasajeros que suben al inicio del intervalo $[t, t+1]$, en la dirección f
$w_{t,f}$	Pasajeros que esperan al inicio del intervalo $[t, t+1]$, en la dirección f

Table 1: Descripción de conjuntos, parámetros y variables de decisión

3.1. Modelo de programación lineal entero mixto basado en restricciones de flujo

El modelo propuesto aborda la optimización de la frecuencia de despachos en rutas de buses de un sistema de transporte público, con el objetivo de minimizar el numero de pasajeros en espera en el sistema y garantizar un uso eficiente de la flota disponible. Basado en un enfoque de flujo temporal en redes, este modelo emplea una estructura de nodos e intervalos de tiempo para representar

los despachos de buses en diferentes franjas horarias y direcciones de trayecto. Cada restricción del modelo responde a una condición operacional crítica, desde la sincronización de los intervalos mínimos y máximos de despacho hasta la gestión de la capacidad y la acumulación de pasajeros en espera. La formulación matemática aquí presentada permite automatizar la programación de viajes, ajustándola dinámicamente a las fluctuaciones en la demanda y optimizando los recursos del operador bajo condiciones variables de carga y tiempo de operación. A continuación se presenta la formulación determinista, que corresponde a un modelo de programación linear entero mixto, el cual puede ser resuelto por medio de algoritmos que garantizan la obtención de soluciones optimas.

3.1.1. Función Objetivo

$$\min z = \sum_{t \in T} \sum_{f \in F} w_{t,f} \tag{1}$$

La función objetivo (1) minimiza el número total de pasajeros en espera $w_{t,f}$ durante todo el periodo de análisis T, tanto para los caminos de ida y de vuelta que son representados por el conjunto F. Esto implica que el modelo busca despachar buses de manera a reducir el tiempo de espera de los usuarios.

La anterior función objetivo esta sujeta a la siguientes restricciones:

3.1.2. Restricción de Intervalos de Despacho

$$h_{\min} \leq \sum_{t \in T} \left(\alpha(t-1) \cdot x^t_{i+1,j,f} - \sum_{t \in T} \alpha(t-1) \cdot x^t_{i,j,f} \right) \leq h_{\max} \tag{2}$$

El conjunto de restricciones (2) garantiza que el intervalo entre despachos de buses esté dentro de un rango aceptable, determinado por los parámetros $h_{\min}$ y $h_{\max}$. Este intervalo asegura que los buses se despachen ni demasiado pronto (evitando una frecuencia innecesariamente alta) ni demasiado tarde (para no aumentar la espera de los pasajeros). Aquí, $x^t_{i,j,f}$ es una variable binaria que indica si el bus i en el servicio j en el intervalo de tiempo t y dirección f es despachado.

3.1.3. Restricción de Tiempo de Viaje

$$\sum_{t \in T} \left(\alpha \cdot (t-1) \cdot x^t_{i,j+1,f} - \sum_{t \in T} \alpha \cdot (t-1) \cdot x^t_{i,j,f} \right) \geq t_{\text{rut}} + t_{\min}, \tag{3}$$

(3) establece que el tiempo entre el inicio de dos servicios consecutivos realizados por el mismo bus (entre j y $j+1$) debe ser al menos igual a la suma del tiempo de la ruta t_{rut} y el tiempo de maniobra $t_{\min}$. Esto garantiza que cada bus tenga suficiente tiempo para completar un viaje y realizar las maniobras necesarias antes de su siguiente despacho.

3.1.4. Restricción de Máximo de Despachos por Intervalo (Eq.(4))

$$\sum_{t \in T} x^t_{i,j,f} \leq 1, \quad \forall f \in F, \forall t \in T, \forall i \in B_f, \forall j \in S_f \tag{4}$$

Aquí, (4) limita el número de despachos en un intervalo t a uno solo. Esto significa que en cada intervalo de tiempo solo puede despacharse un bus por dirección f, lo que es esencial para evitar congestión en los intervalos de salida.

3.1.5. Restricción de Máximo de Despachos por servicio (Eq.(4))

$$\sum_{i \in B_f} \sum_{j \in S_f} x^t_{i,j,f} \leq 1, \quad \forall f \in F, \forall t \in T \tag{5}$$

La restricción 7, asegura que sea realizado máximo un despacho por cada bus i y cada servicio j, en la dirección f. Esto es importante para que no se dupliquen despachos en un bus en los distintos servicios que opera.

3.1.6. Restricción de Espera de Pasajeros

$$w_{(t,f)} = w_{(t-1,f)} + \lambda_{(t-1,f)} \cdot \alpha - b_{(t,f)} \tag{6}$$

El conjunto de ecuaciones (6) calculan la cantidad de pasajeros en espera $w_{t,f}$ al inicio del intervalo $[t, t+1]$. Se determina sumando el número de pasajeros en espera en el intervalo anterior $w_{(t-1,f)}$, más los pasajeros que llegan $\lambda_{(t-1,f)} \cdot \alpha$, y restando los que suben al sistema $b_{(t,f)}$. Esta restricción ayuda a modelar la acumulación de pasajeros en función de los despachos y la frecuencia de llegada.

3.1.7. Restricción de Capacidad del Bus

$$b_{(t,f)} \leq \sum_{i \in B_f} x^t_{(i,f)} \cdot C \tag{7}$$

Seguidamente, (7) asegura que el número de pasajeros que aborda en un intervalo de tiempo no supere la capacidad C del bus. Aquí, C representa la capacidad máxima, y esta restricción previene la sobrecarga del sistema.

3.1.8. Restricción de Naturaleza de las Variables

$$w_{(t,f)}, s_{(t,f)} \in \mathbb{Z}^+, \quad x^t_{(i,f)} \in \{0, 1\} \tag{8}$$

Finalmente, (8) esta restricción define la naturaleza de las variables de decisión: los pasajeros en espera y abordo son variables enteras no negativas, mientras que $x^t_{i,j,f}$ es una variable binaria que indica si un bus ha sido despachado o no.

Es importante anotar que el anterior modelo considera que:

- Un numero fijo de vehículos se encuentra asignado a una ruta especifica
- La frecuencia de salida de vehículos de depósito será regular según la franja horaria
- Los vehículos cuentan con un límite de servicios (basado en el tiempo) desde la salida del depósito.
- No es considerada la transferencia entre distintas rutas
- Con el objetivo de reconocer el comportamiento estocástico de la demanda, se definirán tasas de llegada a las estaciones por línea bajo los niveles de capacidad y solicitación actuales.
- Los tiempos de espera serán modelados a nivel de toda la linea bajo la tasa de llegada definida por modelos probabilísticos

La formulación matemática fue implementada en AMPL y Python, donde fueron probadas en instancias de menor cuantía a las del caso de de estudio. En el momento de evaluar las instancias del caso de estudio, el modelo demuestra infactibilidad. Bajo esta premisa, el modelo planteado en 3.1, es solucionado a partir de una metaherusitica, mientras que la formulación es modificada para poder ser implementada en AMPL y Python de manera eficiente. La formulación modificada es la siguiente:

3.2. Modelo de programación lineal entero mixto basado en restricciones de flujo modificado

Para el modelo modificado se sustrae el set de servicios S, se añaden múltiples parámetros y se modifican y añaden restricciones.

3.2.1. Parametros Adicionales

Con respecto a la tabla 1, se añaden los parámetros: t_{bus}, definido como la cantidad mínima de intervalos entre despachos de un mismo bus. $t_{\text{bus'}}$,definido como la cantidad maxima de intervalos entre despachos del mismo bus.t_{buses}, definido como cantidad minima de intervalos entre despachos de dos buses diferentes. $t_{\text{buses'}}$, definido como la cantidad maxima de intervalos entre despachos de dos buses diferentes. Por ultimo, t_{limite}, es definido como el ultimo intervalo donde se puede despachar un bus antes de cumplir con el horario de operación. El calculo de los parámetros fue realizado bajo las siguientes ecuaciones:

$$t_{\text{bus}} = \frac{2 \cdot t_{\text{ruta}} + 2 \cdot t_{\text{min}}}{\alpha} \tag{9}$$

$$t'_{\text{bus}} = \frac{2 \cdot t_{\text{ruta}} + 2 \cdot t_{\text{min}} + 2 \cdot h_{\text{max}}}{\alpha} \tag{10}$$

$$t_{\text{buses}} = \frac{h_{\text{min}}}{\alpha} \tag{11}$$

$$t'_{\text{buses}} = \frac{h_{\text{max}}}{\alpha} \tag{12}$$

$$t_{\text{limite}} = m - \frac{t_{\text{ruta}}}{\alpha} \tag{13}$$

3.2.2. Función Objetivo modelo modificado

$$\min z = \sum_{t \in T} \sum_{f \in F} w_{t,f} \tag{14}$$

La función objetivo del modelo modificado (14) se mantiene con la misma formulación.

3.2.3. Restricción de máximo despacho por intervalo

$$\sum_{i=1}^{n_f} x_{i,t,f} \leq 1, \quad \forall f \in F,\ t \in T \tag{15}$$

La restricción 15, establece que para cada sentido f y para cada intervalo t, máximo se puede asignar un bus.

3.2.4. Restricción de despacho secuencial

$$\sum_{j=1}^{i-1} \sum_{t_j=1}^{t-1} x_{j,t_j,f} \geq x_{i,t,f}, \quad \forall f \in F,\ i \in B_f,\ t \in T,\ i > 0 \tag{16}$$

La restricción 16, establece que si se asigna el bus i en el intervalo t y en el sentido f se debieron asignar todos los buses anteriores a i en algún intervalo menor a t y en el sentido f .

3.2.5. Restricción de cumplimento de tiempo de ruta

$$x_{i,t,f} + x_{i,t+k,f} \leq 1, \quad \forall f \in F,\ i \in B_f,\ t \in T,\ k \in \{1, 2, \ldots, t_{\text{bus}} - 1\},\ t + k < m \tag{17}$$

La restricción 17, establece que no se puede volver a despachar un mismo bus i sin que este haya cumplido por lo menos con la cantidad mínima de intervalos entre despachos de un mismo bus.

3.2.6. Restricciones de intervalo mínimo y máximo

$$\sum_{k=t_{\text{bus}}}^{t'_{\text{bus}}} x_{i,t-k,f} \geq x_{i,t,f}, \quad \forall f \in F,\ i \in B_f,\ t \in T,\ t - t'_{\text{bus}} \geq 0 \tag{18}$$

$$\sum_{k=t_{\text{buses}}}^{t'_{\text{buses}}} x_{i-1,t-k,f} \geq x_{i,t,f}, \quad \forall f \in F,\ t \in T,\ i \in B_f \setminus \{1\},\ t - t'_{\text{buses}} \geq 0 \tag{19}$$

La restricción 18, establece que si un bus i se asigno en un intervalo t en el sentido f, el siguiente despacho de ese mismo bus debe respetar la cantidad mínima y máxima de intervalos entre despachos de un mismo bus en el sentido f. Mientras que, la restricción 19, define que si un bus i se asigno a un intervalo t en el sentido f, el despacho del siguiente bus $i + 1$ debe respetar la cantidad mínima y máxima de intervalos entre despachos de diferentes buses en el sentido f.

3.2.7. Restricción primer despacho

$$\sum_{t=t_{\text{buses}}}^{t'_{\text{buses}}} x_{1,t,f} \geq 1, \quad \forall f \in F \tag{20}$$

$$w_{1f} = \alpha \cdot \lambda_{1f} - b_{1f}, \quad \forall f \in F \tag{21}$$

La restricción 20, establece que el primer bus debe despacharse en un intervalo t que este entre la cantidad mínima y máxima de intervalos entre despachos de diferentes buses Por otro lado, la ecuación 21 refiere al calculo de los pasajeros en espera en el primer intervalo. Después del primer intervalo $[t : 2, Card(T)]$ Se mantiene la restricción 6.

3.2.8. Restricción despacho limite

$$x_{i,t,f} = 0, \quad \forall f \in F,\ t \in \{t_{\text{limite}}, \ldots, m\},\ i \in B_f \tag{22}$$

La restricción 22, establece que no se puede despachar ningún bus después del intervalo limite.

4. Metodología

Con el objetivo de acelerar el proceso del solver y llegar a una solución exacta se planteo la formulación modificada en la sección 3.2.El modelo planteado en 3.2, se implemento en Python, donde se resolvió con CBC y en AMPL se probo el modelo con algunas variaciones y fue resuelto con Gurobi 11.0.3. Los resultados presentados en 5 fueron los obtenidos con la modelación en python y con el solver CBC. A su vez,en 4.1 se desarrollo una Metaheurística para mejor el tiempo computacional dado por la solución exacta. Mientras que para la formulación inicial (sección 3.1) se desarrollo una Heuristica que se encuentra detallada en A.1 y dado a que no se evaluaron los resultados de esta en detalle fue apartada en Anexos.

4.1. Aproximación MetaHeurística

Inicialmente, resolvimos el modelo planteado utilizando la librería PuLP con el solucionador CBC. Sin embargo, en algunas instancias, el tiempo de ejecución para encontrar la asignación óptima alcanzaba los treinta minutos. Por esta razón, decidimos desarrollar una heurística que encuentra soluciones factibles en un tiempo considerablemente menor y muy cercanas a la solución óptima. La heurística identifica los posibles intervalos factibles para cada bus y asigna el bus a todos los intervalos encontrados sin evaluar si existe una asignación mejor. Este enfoque permite construir rápidamente una solución inicial que se mantiene sin ajustes adicionales a lo largo de la ejecución del algoritmo, optimizando así el tiempo de resolución.

Algorithm 1 Metaheurística para la Creación de una Solución Factible

Parámetros: Algoritmo Constructivo: Creación de una solución factible

Input:

α: Duración de cada intervalo,
t_{ruta}: Tiempo de ruta, $t_{\min}$: Tiempo de maniobra,
$h_{\min}$, $h_{\max}$: Intervalos mínimo y máximo entre servicios,
C: Capacidad del bus, m: Número de intervalos,
λ_{tf}: Tasa de llegada de pasajeros por intervalo para cada sentido,
n_f: Número de buses por sentido,
w_{tf}: Matriz de pasajeros esperando,
b_{tf}: Matriz de pasajeros abordando,
llegadas_{tf}: Matriz de pasajeros llegando.

Salida: Mejor objetivo (`mejor_objetivo`): Número total de personas esperando.

for $f \in \{0, 1, \ldots, F-1\}$ **do**
  for $i \in \{0, 1, \ldots, n_f - 1\}$ **do**
    $\text{rcl} \leftarrow \text{generar\_rcl}(i, f)$
for $f \in \{0, 1, \ldots, F-1\}$ **do**
  for $t \in \{0, 1, \ldots, m-1\}$ **do**
    $b[t, f] = \min\left(\sum_{i=0}^{n_f-1} x[i, f, t] \times C, w[t, f]\right)$
    $w[t, f] \mathrel{-}= b[t, f]$
    $w[t+1, f] = w[t, f] + \text{llegadas}[t+1, f] - b[t+1, f]$
return $\sum_{t=0}^{m-1} \sum_{f=0}^{F-1} w[t, f]$

Algorithm 2 Generar RCL: Lista de Intervalos Factibles para Asignar un Bus

Input: i: Índice del bus, f: Índice del sentido
Output: rcl: Lista con todos los intervalos factibles donde se puede asignar el bus i en el sentido f

$rcl \leftarrow []$
for $t \in \{0, 1, \ldots, m-1\}$ **do**
  $x[i, f, t] \leftarrow 1$
  if $\sum x[:, f, t] > 1$ **then**
    $x[i, f, t] \leftarrow 0$
  $tiempo_{\min} \leftarrow \text{int}\left(\frac{2t_{\text{ruta}}+2t_{\min}}{\alpha}\right)$
  $tiempo_{\max} \leftarrow \text{int}\left(\frac{2t_{\text{ruta}}+2t_{\min}+2h_{\max}}{\alpha}\right)$
  if $t - tiempo_{\max} \geq 0$ **and** $\sum_{k=tiempo_{\min}}^{tiempo_{\max}+1} x[i, f, t-k] < x[i, f, t]$ **then**
    $x[i, f, t] \leftarrow 0$
    continue
  if $t \geq 1$ **and** $t < tiempo_{\min}$ **and** $\sum_{k=0}^{t+1} x[i, f, k] > 1$ **then**
    $x[i, f, t] \leftarrow 0$
    continue
  if $t \geq tiempo_{\min}$ **and** $\sum_{k=0}^{tiempo_{\min}} x[i, f, t-k] > 1$ **then**
    $x[i, f, t] \leftarrow 0$
    continue
  $min_{\text{intervalos}} \leftarrow \text{int}\left(\frac{h_{\min}}{\alpha}\right)$
  $max_{\text{intervalos}} \leftarrow \text{int}\left(\frac{h_{\max}}{\alpha}\right)$
  if $i \geq 1$ **and** $t - max_{\text{intervalos}} \geq 0$ **and** $\sum_{k=min_{\text{intervalos}}}^{max_{\text{intervalos}}+1} x[i-1, f, t-k] < x[i, f, t]$ **then**
    $x[i, f, t] \leftarrow 0$
    continue
  if $t \geq 0$ **and** $t < \text{int}\left(\frac{h_{\min}}{\alpha}\right)$ **and** $i = 0$ **then**
    $x[i, f, t] \leftarrow 0$
    continue
  $t_{\text{ruta\_intervalos}} \leftarrow \text{int}\left(\frac{t_{\text{ruta}}}{\alpha}\right)$
  $t_{\text{limite}} \leftarrow m - t_{\text{ruta\_intervalos}}$
  if $t \geq t_{\text{limite}}$ **and** $t < m$ **then**
    $x[i, f, t] \leftarrow 0$
    continue
  rcl.append(t)
return rcl

5. Resultados numéricos

5.1. Información del caso

El caso de estudio esta enmarcado en Metrosinú el sistema estratégico de transporte publico de Montería. La empresa operadora Movisinú brindo los datos de ventas del sistema para 21 rutas. A partir de esto, recibimos un modelo de predicción de demanda donde se proyecta con series de tiempo en la herramienta de predicción Prophet. Dada la calidad y cantidad de información por ruta, se evalúan unicamente dos rutas: Pradera 27 y Panzenú. La demanda proyectada es dividida en 96 intervalos de 10 minutos cada uno los cuales cubren todo el horario operativo de las rutas.También, asumimos el mismo valor valor de la demanda para el sentido 1 y 2.

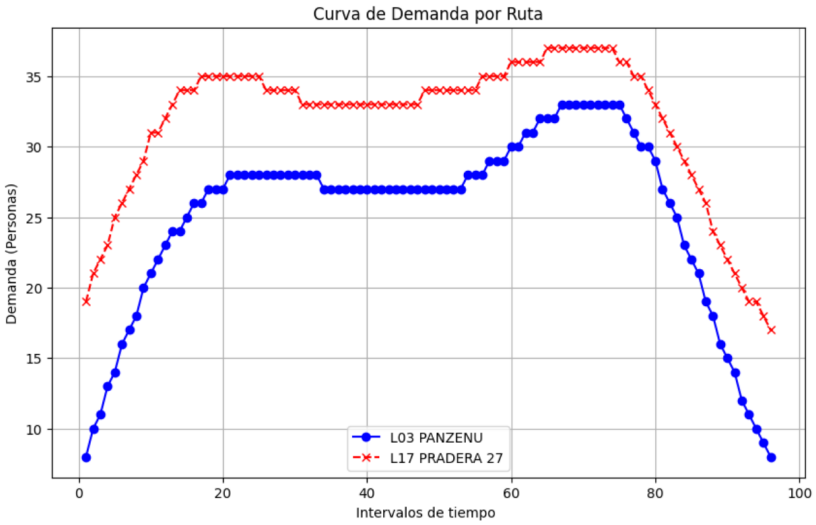

Figure 1: Curva de demanda por ruta

Para definir los parámetros del modelo, contamos con información proporcionada por Movisinú sobre la flota de cada ruta, e información de duración de la misma obtenida de Moovit. Los parámetros definidos son los siguientes:
Parametros Generales:

- Numero de intervalos (m) : 96
- Duración de Intervalo (α): 10 min
- Intervalo Mínimo: 5 min
- Intervalo Máximo: 15 min
- Capacidad del bus (C): 45

Pradera 27:

- Flota (B): 16
- Tiempo de ruta : 40

Panzenu:

- Flota (B): 14
- Tiempo de ruta : 40

5.2. Resultados experimentales

Se definieron dos escenarios para evaluar la eficiencia del modelo en las lineas estudiada. El primero donde la demanda por cada sentido sea exactamente la misma, y el segundo donde esta demanda varié entre las dos lineas.

5.2.1. Escenario 1: Demanda equivalente

Para el primer escenario se evaluó el despacho de los buses con una misma demanda por cada sentido. A partir de esto, se definieron los intervalos de despacho para cada uno de los buses. La primera observación bajo la ruta pradera 27, es que se establecen intervalos regulares de despacho. Se despachan buses consecutivos por cada intervalo, y se hacen despachos del mismo bus cada 9 intervalos por sentido. También, no son utilizados la totalidad de los buses de la ruta. El mismo caso se presenta en la ruta panzenu (3 fig.(2)

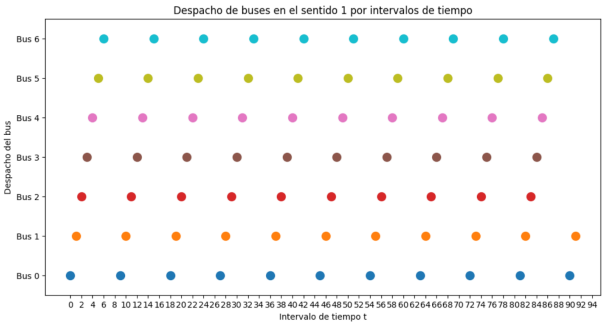

Figure 2: Despacho de buses - Pradera 27

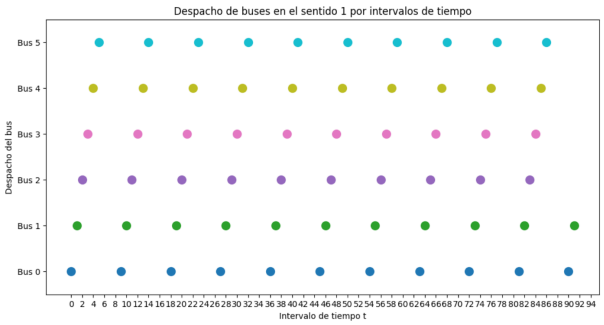

Figure 3: Despacho de buses- Panzenu

Bajo estas condiciones, Los pasajeros que abordan el bus se encuentran definidos por los momentos de despacho de los buses, por lo que en la mayoría de los casos se carga a la capacidad máxima del bus. Entre las dos rutas analizadas se presenta un comportamiento similar, sin embargo, dada la ligera reducción en demanda de la linea panzenu se evidencian diferencias en cuanto a magnitud. A su vez, los pasajeros en espera, se cargan a la tasa Lambda, donde por cada intervalo que esperan el despacho de un bus incrementan y se descargan con el despacho del bus. Nuevamente, dado que la curva de demanda de las rutas es similar se presentan comportamientos con el mismo regimen.

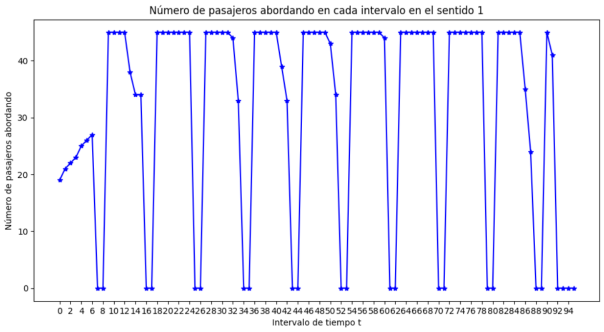

Figure 4: Pasajeros Abordando - Pradera 27

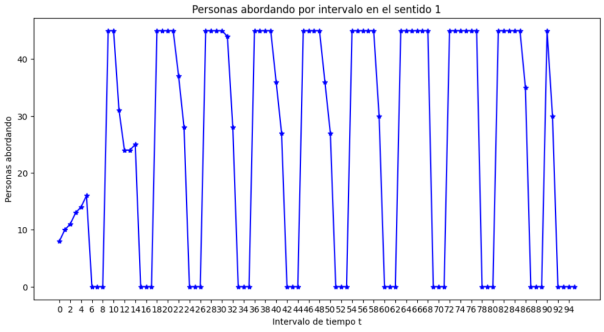

Figure 5: Pasajeros abordando- Panzenu

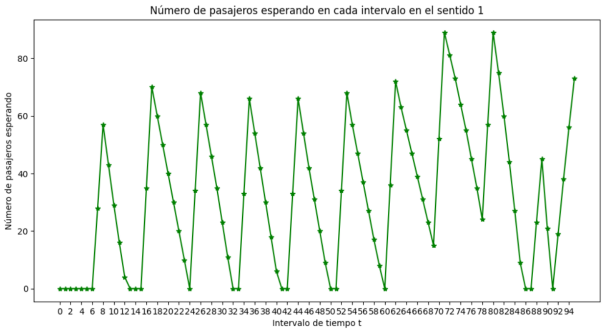

Figure 6: Pasajeros en espera - Pradera 27

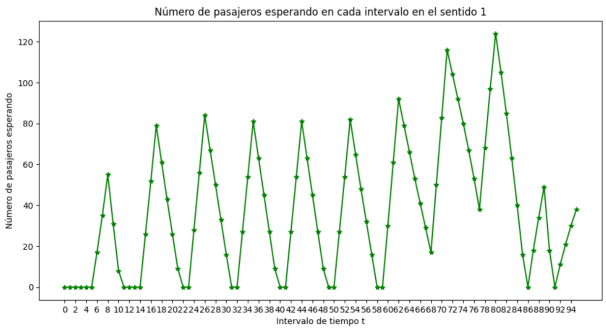

Figure 7: Pasajeros en espera - Panzenu

Las estadísticas computacionales de evaluar los dos sentidos bajo la misma demanda son las siguientes:

c = 45, h_min = 5m, h_max = 15m

n1	n2	w1	w2	CPU time
1	15	2562	73	71.75s
2	14	2135	73	79.13s
3	13	1707	73	137.35s
4	12	1277	73	85.63s
5	11	846	73	96.87s
6	10	396	73	88.57s
7	9	73	73	166.21s
8	8	73	73	1265.08s
7	7	73	73	394.17s
6	6	396	396	40.91s

Figure 8: Estadísticas computacionales - Pradera 27

c = 45, h_min = 5m, h_max = 15m

n1	n2	w1	w2	CPU time
1	13	1949	38	184.82s
2	12	1499	38	129.99s
3	11	1083	38	161.13s
4	10	670	38	246.3s
5	9	286	38	172.19s
6	8	38	38	806.86s
7	7	38	38	1608.28s
6	6	38	38	1935.26s
5	5	286	286	183.13s

Figure 9: Estadísticas computacionales - Panzenu

A continuación, se realizó un análisis de sensibilidad sobre el número de pasajeros esperando y el tiempo de ejecución del solucionador. En este caso, la variable que se decidió modificar fue el número de vehículos disponibles para cada sentido. Para la ruta de Pradera ya que se tienen 16 vehículos disponibles, comenzamos asignando un vehículo para el primer sentido y los demás para el segundo, aumentando progresivamente hasta que la cantidad de vehículos fuer equivalente. En la tabla resultante, se observa que el número de pasajeros esperando en el último intervalo en el sentido 2 se mantiene en 73 mientras la disponibilidad de vehículos se encuentre entre 15 y 7. Por otro lado, en el sentido 1, el número de pasajeros esperando en el último intervalo disminuye a medida que se incrementa el número de vehículos, alcanzando un mínimo de 73 pasajeros con 7 u 8 vehículos asignados. Dado que el objetivo es minimizar la cantidad total de pasajeros esperando en todos los intervalos en ambos sentidos, la solución óptima consiste en asignar 8 vehículos a cada sentido de la línea con un tiempo de ejecución aproximado de 21 minutos. Para la ruta de Panzenu, disponemos de un total de 14 vehículos, aplicando la misma estrategia de asignación utilizada en la ruta de Pradera. En la tabla resultante, se observa que el número de pasajeros esperando en el último intervalo en el sentido 2 se mantiene en 38 mientras la disponibilidad de vehículos se encuentra entre 13 y 6. En el sentido 1, el número de pasajeros esperando en el último intervalo disminuye conforme aumenta el número de vehículos asignados, alcanzando un mínimo de 38 pasajeros con 6 o 7 vehículos. Dado que el objetivo es minimizar la cantidad total de pasajeros esperando en todos los intervalos en ambos sentidos, la solución óptima consiste en asignar 7 vehículos a cada sentido de la línea. Mientras que para la ruta de Pradera el tiempo de CPU fue de aproximadamente 21 minutos, en la ruta de Panzenu el tiempo de CPU se incrementó a 27 minutos. Aunque la demanda en Panzenu es menor que en Pradera, el número de vehículos disponibles en el óptimo se redujo de

16 a 14, manteniéndose los demás parámetros. Esto implica que el solucionador debe realizar más iteraciones para encontrar el óptimo, lo que aumenta el tiempo de ejecución.

5.2.2. Escenario 2: Demanda diferenciada

En el segundo escenario se tomo la demanda del sentido 1 bajo las mismas condiciones del escenario 1 (5.2.1), sin embargo, para el sentido 2, la demanda fue alterada a propósito para evaluar el cambio en los despachos. Esta cambio consistió unicamente en duplicar la demanda del segundo tercio de los intervalos

Para la linea Pradera 27, las primeras observaciones son la utilización de mas buses desde el inicio de los intervalos para el sentido 2, donde los intervalos de despacho en el mismo bus i se reducen. El sentido 1 continua con el mismo comportamiento previo, contamos con los siguientes regímenes de despachos:

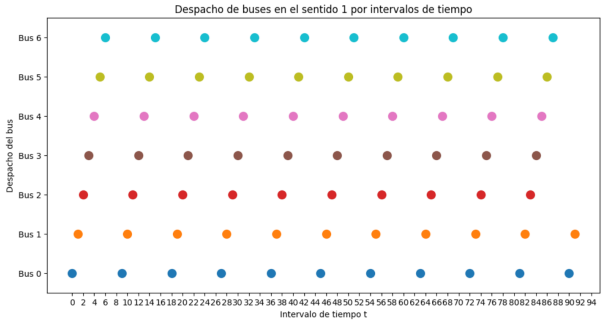

Figure 10: Despachos en sentido 1 - Pradera 27

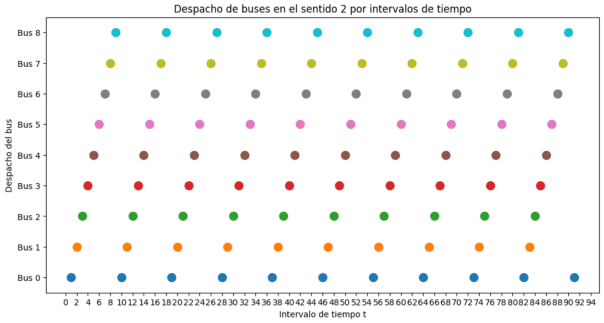

Figure 11: Despachos en sentido 2 - Pradera 27

Los pasajeros esperando por sentido cambian considerablemente. En el sentido 1 se presenta el mismo comportamiento presentado anteriormente, sin embargo, en el sentido 2, dado que se presenta la operación de dos buses mas, los pasajeros en espera de los intervalos iniciales son nulos, con un comportamiento de demanda igual al sentido 1. Por otro lado, cuando se duplica la demanda la operación del sistema se ve desbordada y obtenemos cifras inconsistentes del numero de pasajeros esperando. Algo similar ocurre en los pasajeros abordando, en el sentido 1 contamos con el mismo comportamiento, mientras que en el sentido 2, los pasajeros abordando incrementan conforme al aumento de demanda, sin embargo, al duplicar la demanda se aborda a capacidad máxima hasta el final del periodo.

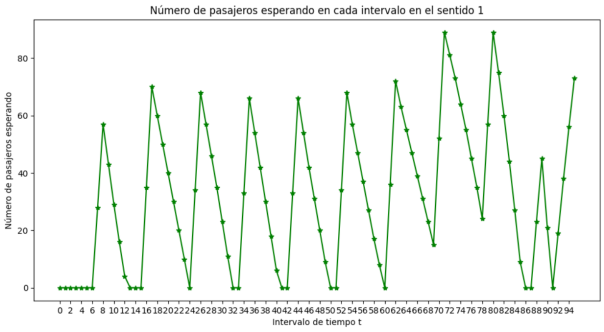

Figure 12: Pasajeros en espera sentido 1 - Pradera 27

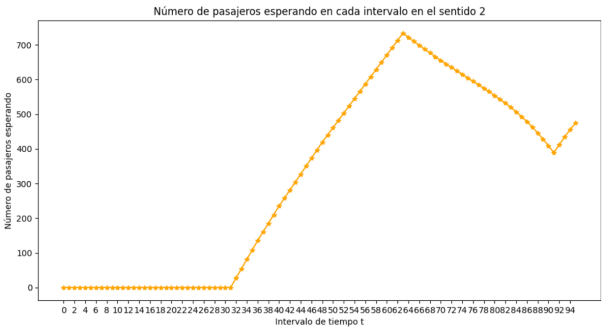

Figure 13: Pasajeros en espera sentido 2 - Pradera 27

Los abordajes al sistema también cuentan con diferencias notorias, donde desde los intervalos

que duplican la demanda hasta m - 2, cuentan con la capacidad maxima de abordaje.

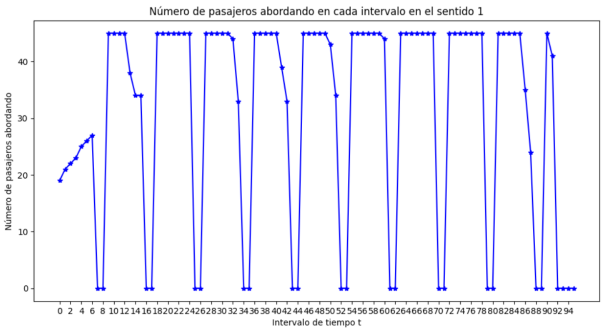

Figure 14: Pasajeros abordando sentido 1 - Pradera 27

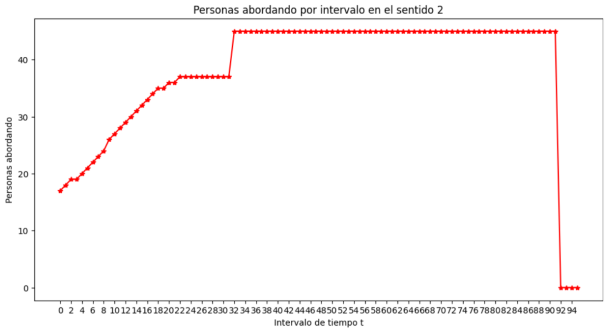

Figure 15: Pasajeros abordando sentido 2 - Pradera 27

c = 45, h_min = 5m, h_max = 15m

n1	n2	w1	w2	CPU time
1	15	2562	474	102.39s
2	14	2135	474	95.66s
3	13	1707	474	126.07s
4	12	1277	474	97.92s
5	11	846	474	76.25s
6	10	396	474	49.59s
7	9	73	474	244.11s
8	8	73	789	300.19s

Figure 16: Estadisticas del modelo - Pradera 27

En la tabla resultante 16, se observa que el número de pasajeros esperando en el último intervalo en el sentido 2 se mantiene en 474 cuando la disponibilidad de vehículos varía entre 15 y 9. En el sentido 1, el número de pasajeros en espera disminuye conforme aumenta el número de vehículos asignados, alcanzando un mínimo de 73 pasajeros con 7 y 8 vehículos. Dado que el objetivo es minimizar el número total de pasajeros esperando en todos los intervalos en ambos sentidos, la asignación óptima de vehículos es de 7 para el sentido 1 y de 9 para el sentido 2. A diferencia del escenario en el que la demanda de ambos sentidos es igual, el tiempo de ejecución se redujo a aproximadamente cuatro minutos.

Para Panzenu, la segunda ruta de análisis, se presentan las gráficas de despacho, pasajeros en espera y pasajeros abordando. El comportamiento de las variables de decisión del modelo son similares al escenario diferenciado de pradera 27, sin embargo, dado a que se tiene una demanda menor, la acumulación de pasajeros en espera se descarga a una tasa mayor que el escenario de pradera 27. A su vez, los pasajeros abordando también cuentan con un comportamiento similar al sentido 1.

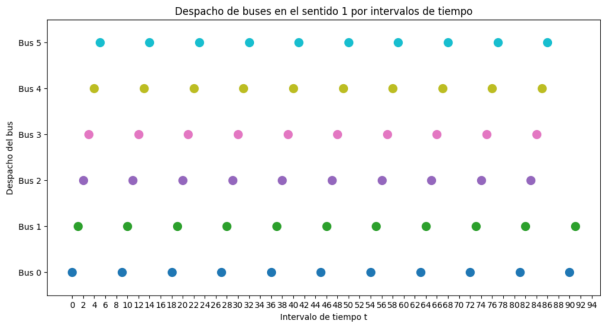

Figure 17: Despachos en sentido 1 - Panzenu

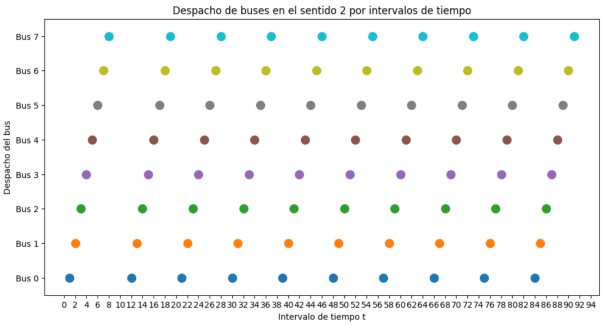

Figure 18: Despachos en sentido 2 - Panzenu

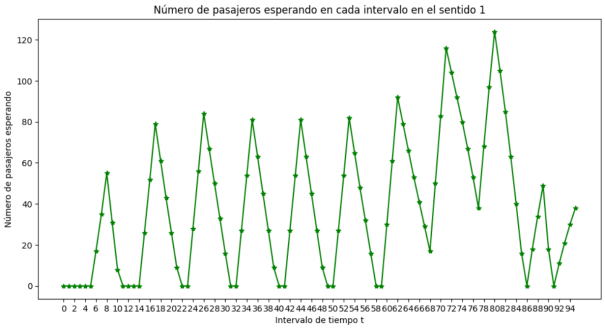

Figure 19: Pasajeros en espera sentido 1 - Panzenu

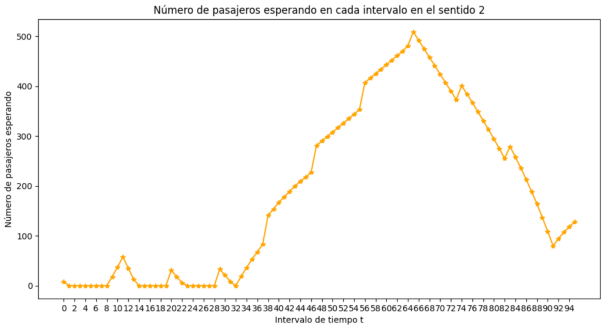

Figure 20: Pasajeros en espera sentido 2 - panzenu

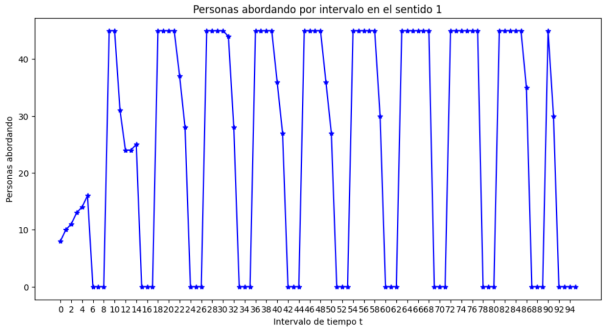

Figure 21: Pasajeros abordando sentido 1 - Panzenu

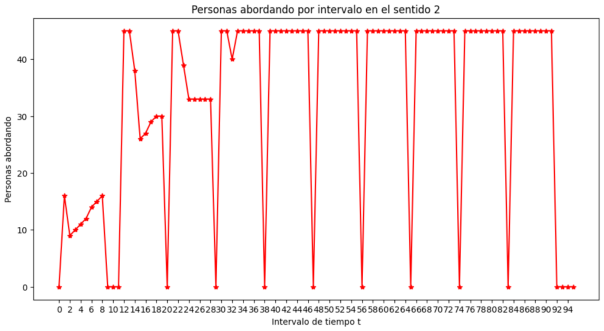

Figure 22: Pasajeros abordando sentido 2 - Panzenu

c = 45, h_min = 5m, h_max = 15m

n1	n2	w1	w2	CPU time
1	13	1949	42	129.91s
2	12	1499	42	54.89s
3	11	1083	42	96.74s
4	10	670	42	154.14s
5	9	286	42	104.53s
6	8	38	42	434.17s
7	7	38	455	810.90s
8	6	38	811	254.57s

Figure 23: Estadísticas del modelo - Panzenu

En la tabla resultante 23, se observa que el número de pasajeros esperando en el último intervalo en el sentido 2 se mantiene en 42 cuando la disponibilidad de vehículos varía entre 13 y 8. En el sentido 1, el número de pasajeros en espera disminuye conforme aumenta el número de vehículos asignados, alcanzando un mínimo de 38 pasajeros con 8,7 o 6 vehículos. Dado que el objetivo es minimizar el número total de pasajeros esperando en todos los intervalos en ambos sentidos, la asignación óptima de vehículos es de 6 para el sentido 1 y de 8 para el sentido 2. A diferencia del escenario en el que la demanda de ambos sentidos es igual, el tiempo de ejecución se redujo a aproximadamente siete minutos.

5.2.3. Estadísticas Metaheurística

Con el objetivo de evaluar el desempeño de la metaheurística desarrollada, se presentan las tablas de estadísticas computacionales.

c = 45, h_min = 5m, h_max = 15m

n1	n2	w1	w2	CPU time
1	15	2583	73	0.002s
2	14	2112	73	0.003s
3	13	1685	73	0.004s
4	12	1257	73	0.003s
5	11	827	73	0.003s
6	10	396	73	0.002s
7	9	73	73	0.002s
8	8	73	73	0.002s
7	7	73	73	0.003s
6	6	396	396	0.003s

Figure 24: Estadísticas de la Metaheurística- Pradera

c = 45, h_min = 5m, h_max = 15m

n1	n2	w1	w2	CPU time
1	13	1954	38	0.003s
2	12	1499	38	0.003s
3	11	1083	38	0.003s
4	10	670	38	0.004s
5	9	286	38	0.005s
6	8	38	38	0.005s
7	7	38	38	0.002s
6	6	38	38	0.007s
5	5	286	286	0.008s

Figure 25: Estadísticas de la Metaheurística - Panzenu

Para la línea Pradera fig(24), la heurística encuentra la asignación óptima en un tiempo de 0.002 segundos. Sin embargo, en otros escenarios, su rendimiento difiere del algoritmo exacto. Por ejemplo, al asignar 1 vehículo al sentido 1 y 15 vehículos al sentido 2, el último intervalo en el sentido 1 tiene 2,583 personas esperando, un valor superior al óptimo en ese escenario, aunque aún cumple con las restricciones del problema. En los siguientes 4 escenarios, el número de pasajeros esperando en el último intervalo en el sentido 1 es menor que el óptimo encontrado por el algoritmo exacto. Sin embargo, el total de pasajeros esperando en todos los intervalos en ambos sentidos es mayor que en el óptimo para esos escenarios. Para los últimos escenarios, tanto la heurística como el algoritmo exacto logran los mismos intervalos de despacho y el mismo número total de pasajeros esperando en todos los intervalos en ambos sentidos.

Para la línea Panzenu fig(25), la heurística encuentra la asignación óptima en un tiempo de 0.002 segundos. Sin embargo, en otros escenarios, su rendimiento difiere del algoritmo exacto. Por ejemplo, al asignar 1 vehículo al sentido 1 y 13 vehículos al sentido 2, el último intervalo en el sentido 1 tiene 1954 personas esperando, un valor superior al óptimo en ese escenario, aunque aún cumple con las restricciones del problema. En los siguientes 4 escenarios, el número de pasajeros esperando en el último intervalo en el sentido 1 es igual que el óptimo encontrado por el algoritmo exacto. Sin embargo, el total de pasajeros esperando en todos los intervalos en ambos sentidos es mayor que en el óptimo para esos escenarios. Para los últimos escenarios, tanto la heurística como el algoritmo exacto logran los mismos intervalos de despacho y el mismo número total de pasajeros esperando en todos los intervalos en ambos sentidos.

6. Discusión

7. Conclusión

A. Nomenclature

Referencias

References

[1] Stephan H., Avishai C., 2014. Public transport vehicle scheduling featuring multiple vehicle types. Transportation Research Part B. 67, 129–143.

[2] Samuela C. , Antonio F., Laura G., Leopoldo G., Giuliano V., 2019. A matheuristic for integrated timetabling and vehicle scheduling. Transportation Research Part B. 127, 99–124.

[3] Gilbert L., Francisco O., Miguel P., Justo P., 2017. Multi-objective integration of timetables, vehicle schedules and user routings in a transit network. Transportation Research Part B. 98, 94–112.

[4] Valérie G., Jin-Kao H., 2010. Transit network timetabling and vehicle assignment for regulating authorities. Computers and Industrial Engineering. 59, 16–23.

[5] Omar I., Yasmin R., 2012. Synchronization of bus timetabling. Transportation Research Part B. 46, 599–614.

[6] James C., Kanticha K., Yu-Ting H., Hua-Yen W., 2019. Models and a solution algorithm for planning transfer synchronization of bus timetables. Transportation Research Part E. 131, 247–266.

[7] Mathias M., Anita S., 2009. Integrating line planning, timetabling, and vehicle scheduling: a customer-oriented heuristic. Springerlink.

[8] Laurent, B., & Hao, J. K. (2008). Simultaneous vehicle and crew scheduling for extra urban transports. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5027 LNAI. https://doi.org/10.1007/978-3-540-69052-8_49

[9] André de P., Robin L., 2001. Optimal timetables for public transportation. Transportation Research Part B. 35, 789–813.

[10] LIU Z., SHEN J., 2007. Regional Bus Operation Bi-level Programming Model Integrating Timetabling and Vehicle Scheduling. Systems Engineering - Theory and Practice. 27, 135–141.

[11] Avishai C., 2011. Optimal Multi-Vehicle Type Transit Timetabling and Vehicle Scheduling. Procedia Social and Behavioral Sciences. 20, 19–30.

[12] Hua-Yan S., Hai-Jun H., Wen-X., 2020. Bus timetabling considering passenger satisfaction: An empirical study in Beijing . Computers & Industrial Engineering . 135, 1155–1166.

[13] Huayan S., Yanping L.,Haijun H., and Renyong G., 2019. Vehicle Scheduling Optimization considering the Passenger Waiting Cost. Hindawi, Journal of Advanced Transportation, 1–13.

[14] Ali H., Mohamadreza B., Kun-Hung C., 2001. A comparative analysis of bus transit . Transportation Research Part B. 37, 301–322.

[15] João F., Evelien H., Roberto R., Allan L., 2018. A matheuristic for transfer synchronization through integrated timetabling and vehicle scheduling . Transportation Research Part B - Q1 . 109, 128–149.

[16] Ibarra-Rojas, O. J., Delgado, F., Giesen, R., Muñoz, J. C. (2015). Planning, operation, and control of bus transport systems: A literature review. In Transportation Research Part B: Methodological (Vol. 77). https://doi.org/10.1016/j.trb.2015.03.002

[17] Meng, G., Lai, Y., Yang, F. (2020). An optimal bus scheduling model based on mixed-integer linear programming. https://doi.org/10.1109/ITAIC49862.2020.9338826

[18] P. da Cunha, J. M., Rodríguez Vignoli, J. (2009). Crecimiento urbano y movilidad en América Latina. Revista Latinoamericana de Población, 3(4–5). https://doi.org/10.31406/relap2009.v3.i1.n4-5.1

[19] Sharma, S., Bhattacharya, S., Kiran, D., Hu, B., Prandtstetter, M., Azzopardi, B. (2023). Optimizing the Scheduling of Electrified Public Transport System in Malta †. Energies, 16(13). https://doi.org/10.3390/en16135073

[20] Zhang, J., Li, W., Qiu, F. (2015). Optimizing Single-Depot Vehicle Scheduling Problem: Fixed-Interval Model and Algorithm. Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, 19(3). https://doi.org/10.1080/15472450.2013.836930

[21] Kulkarni, S., Krishnamoorthy, M., Ranade, A., Ernst, A. T., Patil, R. (2018). A new formulation and a column generation-based heuristic for the multiple depot vehicle scheduling problem. Transportation Research Part B: Methodological, 118. https://doi.org/10.1016/j.trb.2018.11.007

[22] Davatgari, A., Cokyasar, T., Verbas, O., Mohammadian, A. (Kouros). (2024). Heuristic solutions to the single depot electric vehicle scheduling problem with next day operability constraints. Transportation Research Part C: Emerging Technologies, 163, 104656. https://doi.org/10.1016/J.TRC.2024.104656

[23] Teng, J., Jin, S., Lai, X., Chen, S. (2015). Vehicle-scheduling model for operation based on single-depot. Mathematical Problems in Engineering, 2015. https://doi.org/10.1155/2015/506794

[24] Desfontaines, L., & Desaulniers, G. (2018). Multiple depot vehicle scheduling with controlled trip shifting. Transportation Research Part B: Methodological, 113. https://doi.org/10.1016/j.trb.2018.05.011

[25] Barrena, E., Canca, D., Coelho, L. C., & Laporte, G. (2014). Exact formulations and algorithm for the train timetabling problem with dynamic demand. Computers and Operations Research, 44. https://doi.org/10.1016/j.cor.2013.11.003

[26] Hassannayebi, E., & Zegordi, S. H. (2017). Variable and adaptive neighbourhood search algorithms for rail rapid transit timetabling problem. Computers and Operations Research, 78. https://doi.org/10.1016/j.cor.2015.12.011

Anexos

A.1. Heuristica desarrollada para el modelo inicial

GRASP (Greedy Randomized Adaptive Search Procedure) es un enfoque iterativo para resolver problemas de optimización combinatoria. En cada iteración, consta de dos fases: construcción A.1.1 y búsqueda local A.1.2.

A.1.1. Fase de construcción

La solución inicial se genera de manera iterativa, agregando un elemento en cada paso. La elección del próximo elemento se basa en una función greedy, que evalúa el beneficio inmediato de añadir ese elemento, sin considerar cómo afectará a las elecciones futuras. Esto significa que la estrategia es miope, ya que optimiza solo en función de la iteración actual.El carácter adaptativo del proceso implica que, tras cada selección, los beneficios de los elementos restantes se actualizan para reflejar los cambios provocados por las elecciones previas. Así, la solución se construye de forma dinámica, ajustando continuamente los beneficios asociados a los elementos no seleccionados.

En cada paso, se elabora una lista de candidatos con los elementos que pueden añadirse a la solución en ese momento. Esta lista se ordena según la función greedy, y a partir de ahí se crea una lista restringida de candidatos con aquellos elementos que ofrecen mayores beneficios en términos del criterio de optimización. A diferencia de un enfoque puramente determinista, GRASP selecciona aleatoriamente un elemento de esta lista restringida, lo que introduce un factor de aleatoriedad en la construcción de la solución.

A.1.2. Búsqueda local

La solución obtenida en la fase de construcción no garantiza la optimalidad local, por lo que se aplica un procedimiento de búsqueda local para mejorarla. En esta fase, se explora el entorno de la solución actual, que es el conjunto de soluciones que se pueden obtener mediante pequeños ajustes (movimientos) en la solución inicial. Se elige una nueva solución en este entorno, buscando mejorar el valor de la solución de partida. En cada iteración, el algoritmo genera una nueva solución perteneciente al entorno de la solución previa, mediante un movimiento que explora una posible mejora. Esto garantiza que la metodología GRASP no solo se enfoque en la construcción de soluciones iniciales, sino también en su optimización mediante ajustes sucesivos.

A continuación presentaremos los algoritmos referentes a la fase de construcción y búsqueda local aplicados a problema de programación de autobuses.

Algoritmo GRASP

Este primer algoritmo define dos variables, una para la asignación de los servicios y otra para el numero total de pasajeros esperando. Estas variables van a ser actualizadas si el algoritmo encuentra una mejor solución en cada una de las iteraciones.

Algorithm 3 GRASP

Parámetros: Max_iter: Número de iteraciones,
Algoritmo Constructivo: Algoritmo para la creación de soluciones factibles,
Busqueda Local: Algoritmo de Búsqueda Local

Input: α: Duración de cada intervalo,
t_ruta: Tiempo de ruta, t_min: Tiempo de maniobra,
h_min: Intervalo mínimo entre servicios, h_max: Intervalo máximo entre servicios,
C: Capacidad del bus, λ_tf: Tasa de llegada de pasajeros por intervalo para cada sentido

Output: $mejor\_solucion$: Lista de tuplas con los servicios asignados,
$mejor\_objetivo$: Número total de personas esperando

$mejor\_solucion \leftarrow$ None
$mejor\_objetivo \leftarrow$ Número muy grande
for $iteracion \in \{0, 1, 2, \ldots, Max\_iter - 1\}$ **do**
    $lista\_indices \leftarrow []$
    Ejecutar Algoritmo Constructivo
    Ejecutar Algoritmo de Búsqueda Local
    if sumatoria de $w < mejor\_objetivo$ **then**
        $mejor\_objetivo \leftarrow$ Sumatoria de w
        $mejor\_solucion \leftarrow lista\_indices$
return $mejor\_solucion, mejor\_objetivo$

Algoritmo Constructivo

El algoritmo constructivo para cada servicio de un bus en un sentido crea una lista de todos los intervalos donde se puede asignar ese servicio cumpliendo las restricciones del modelo. De esa lista escoge un valor de manera aleatoria y asigna el servicio a ese intervalo, calcula la cantidad de personas que abordan y re calcula el total de personas esperando dado que el servicio j del bus i en el sentido f recoge pasajeros en ese intervalo t. Después de la asignación se ejecuta el algoritmo de búsqueda local que mejore la solución actual.

Algorithm 4 Algoritmo Constructivo

Parámetros: m: Cantidad de intervalos, F: Cantidad de direcciones, α: Longitud del intervalo,
$n_f[f]$: Cantidad de vehículos para la dirección f,
$s_f[f]$: Cantidad de servicios para cada bus en la dirección f

for $f \in \{0, 1, \ldots, F-1\}$ **do**
  for $j \in \{0, 1, \ldots, s_f[f]-1\}$ **do**
    for $i \in \{0, 1, \ldots, n_f[f]-1\}$ **do**
      $rcl \leftarrow$ Construccion RCL(i, f, j)
      if tamaño de rcl es igual a 0 **then**
        continuar con la siguiente iteración
      $t \leftarrow$ Escoger aleatoriamente un elemento de la lista rcl
      $x[i, j, f, t] \leftarrow 1$
      $b[t, f] \leftarrow$ calcular_abordaje(f, t)
      for $t' \in \{0, 1, 2, \ldots, m-1\}$ **do**
        $w[t', f] \leftarrow$ calcular_pasajeros_esperando(t', f)
Ejecutar el algoritmo de búsqueda local

Algoritmo de búsqueda local

El algoritmo inicializa con dos variables que se actualizaran cada vez que el algoritmo encuentre una mejor solución. Como en el algoritmo constructivo definimos el intervalo de un servicio para un bus en una dirección, vamos a iterar sobre todos los posibles intervalos en los que puede estar ese servicio minimizando la cantidad de personas esperando. En un principio el algoritmo quita la asignación anterior, asigna temporalmente el servicio al nuevo intervalo calcula la cantidad de pasajeros que abordarían y la cantidad total de pasajeros esperando, en el caso de que la cantidad de pasajeros sea menor al intervalo anterior, guarda en una de las variables de inicializacion ese intervalo hasta que termina la lista de todos los posibles candidatos. Cuando finalizan las iteraciones si la variable de mejor solución toma un valor distinto a menos 1 quita definitivamente la asignación del intervalo en el que estaba y asigna el servicio al intervalo optimo encontrado, en caso de que la variable de mejor solución tome el valor de menos 1, se conservaría la asignación del algoritmo de construcción.

Algorithm 5 Algoritmo de Búsqueda Local

$mejor\_objetivo \leftarrow$ Sumatoria de w
$mejor\_solucion\_t \leftarrow -1$
for $t\_alternativo \in rcl$ **do**
    $b[t, f] \leftarrow 0$
    $b[t\_alternativo, f] \leftarrow$ calcular_abordaje$(f, t_alternativo)$
    for $t1 \in \{0, 1, 2, \ldots, m-1\}$ **do**
        $w[t1, f] \leftarrow$ calcular_pasajeros_esperando$(t1, f)$
    $nuevo\_objetivo \leftarrow$ Sumatoria de w
    if $nuevo\_objetivo < mejor\_objetivo$ **then**
        $mejor\_objetivo \leftarrow nuevo\_objetivo$
        $mejor\_solucion\_t \leftarrow t\_alternativo$
        $b[t\_alternativo, f] \leftarrow 0$
        for $t1 \in \{0, 1, 2, \ldots, m-1\}$ **do**
            $w[t1, f] \leftarrow$ calcular_pasajeros_esperando$(t1, f)$
    else
        $x[i, j, f, t\_alternativo] \leftarrow 0$
        $b[t\_alternativo, f] \leftarrow$ calcular_abordaje$(f, t_alternativo)$
        for $t1 \in \{0, 1, 2, \ldots, m-1\}$ **do**
            $w[t1, f] \leftarrow$ calcular_pasajeros_esperando$(t1, f)$
if $mejor\_solucion\_t > -1$ **then**
    $x[i, j, f, t] \leftarrow 0$
    $x[i, j, f, mejor\_solucion\_t] \leftarrow 1$
    $b[mejor\_solucion\_t, f] \leftarrow$ calcular_abordaje$(f, mejor_solucion_t)$
    for $t1 \in \{0, 1, 2, \ldots, m-1\}$ **do**
        $w[t1, f] \leftarrow$ calcular_pasajeros_esperando$(t1, f)$
    Agregar a la lista de índices la tupla $(i, j, f, mejor\_solucion\_t)$
else
    Agregar a la lista de índices la tupla (i, j, f, t)