

Universidad
Tecnológica
de Pereira

PROGRAMA DE DOCTORADO EN INGENIERÍAS

TESIS DOCTORAL:

SOLUCIÓN AL PROBLEMA DE ASIGNACIÓN DE CONDUCTORES Y
VEHÍCULOS EN SISTEMAS DE TRANSPORTE PÚBLICO MASIVO DE
PASAJEROS MEDIANTE LA IMPLEMENTACIÓN DE UNA TÉCNICA
MATEHEURÍSTICA DE OPTIMIZACIÓN

ESTUDIANTE: RUBÉN IVÁN BOLAÑOS

DIRECTOR: Ph.D. MAURICIO GRANADA ECHEVERRI

ASESOR EMPRESA: Ph.D. CÉSAR AUGUSTO MARÍN MORENO

FACULTAD DE INGENIERÍAS
DOCTORADO EN INGENIERÍA
UNIVERSIDAD TECNOLÓGICA DE PEREIRA
PEREIRA

2021

SOLUCIÓN AL PROBLEMA DE ASIGNACIÓN DE CONDUCTORES Y
VEHÍCULOS EN SISTEMAS DE TRANSPORTE PÚBLICO MASIVO DE
PASAJEROS MEDIANTE LA IMPLEMENTACIÓN DE UNA TÉCNICA
MATEHEURÍSTICA DE OPTIMIZACIÓN

RUBÉN IVÁN BOLAÑOS

Presentado como requisito
para optar al título de
Doctor en Ingeniería

FACULTAD DE INGENIERÍAS
DOCTORADO EN INGENIERÍA
UNIVERSIDAD TECNOLÓGICA DE PEREIRA
PEREIRA
2021

Año 2021

NOTA DE ACEPTACIÓN

[b]

[t]

[b]

Jurado [t]

[b]

Jurado [t]

Pereira, 9 de agosto de 2021

DEDICATORIA

Dedicada a las dos mujeres más importantes de mi vida:

Mi esposa Andrea Velásquez Hincapié

Mi madre Luz Marina Bolaños

Rubén Iván Bolaños.

AGRADECIMIENTOS

Son innumerables las personas que han hecho parte de este logro, el cual hoy se constituye como uno de los más importantes en mi vida académica, profesional, laboral y personal; a todos los agradezco infinitamente y espero haber llenado las expectativas que cada uno tenía sobre mí.

En primera medida agradezco a Dios por guiar mi camino en todo mi proceso de formación.

A mi esposa Andrea Velásquez, quien han sido mi apoyo incondicional y fuente de inspiración, compañía que empezó desde mi formación profesional y que hoy junto a ella puedo culminar esta etapa doctoral, a ella mis mas profundos agradecimientos por brindarme su amor, darme su tiempo y por comprenderme en cada decisión que tomo.

A mi madre Luz Marina y a mis hermanas Gloria, Laddy y Carolina quienes han sido las mujeres de mi vida y mi motivo para salir adelante cada día, a ellas toda mi gratitud por siempre confiar en mis capacidades. Así mismo, agradezco a mis sobrinos por verme como el ejemplo a seguir y quienes, sin saberlo aún, pusieron mayor valor a este logro, espero haberlos motivado lo suficiente para que sigan mis pasos sobre este maravilloso camino académico que les he podido mostrar.

A Mauricio Granada Echeverri, director de esta tesis por haber creído en mis capacidades y darme la oportunidad de desarrollar esta investigación de manera abierta y libre; agradezco su disposición y aportes al desarrollo de esta tesis y sus contribuciones en todas las etapas de mi formación.

A César Augusto Marín Moreno, mi mentor durante los últimos 6 años, quien ha sabido guiarme y ha compartido toda su experiencia de manera sabia e incondicional, todas nuestras jornadas de trabajo incansable, la disciplina inculcada y discusiones, hoy se ven materializadas en este logro,

que no solo es el fruto de un trabajo académico, sino que es el fruto de una relación personal basada en la lealtad. A César mis más profundo respeto, admiración, agradecimiento y toda mi lealtad; gracias por haber cambiado mi vida.

Al equipo de trabajo que ha hecho posible este logro, Luis Miguel Escobar Falcón quien puso a mi disposición todos sus conocimientos académicos para el desarrollo de esta tesis y su capacidad de transmitir su saber, a Kenny Cárdenas Parra por sus valiosos aportes y permitir que desde el equipo de trabajo lo encamináramos en el mundo de la Investigación de Operaciones y a Doris Amparo Martínez por haber sido mi coequipera durante los últimos 4 años, a ella mil gracias por ser mi soporte en cada una de las actividades laborales del día a día.

Durante este proceso han sido muchas personas las que de una u otra manera me han motivado y alentado para la consecución de la meta trazada, tanto familiares como amigos, de manera especial agradezco a Jhon Janer Valencia y Sebastián Alzate Rusy, quienes siempre me han apoyado y han creído en mi como amigo y como profesional.

A Edouard Batot por sus largas conversaciones académicas, culturales y reflexiones que compartimos en el siempre sorprendente frío de Montreal; gracias por recordarme que siempre debo ser como el agua “Be Water” y a Pedro Pablo Cárdenas Alzate por su apoyo y compañía durante nuestra estadía en Montreal, a ellos dos, infinitas gracias por todo lo compartido, solo me quedan grandes enseñanzas y magníficas experiencias de vida.

Agradecimientos especiales a las instituciones que hicieron parte de esta formación, la Universidad Tecnológica de Pereira mi Alma Mater y a la empresa Integra S.A. que, durante los últimos 6 años, se ha convertido en mi segundo hogar y me ha permitido crecer personal y profesionalmente.

Gracias a Ramón Antonio Toro Pulgarín, gerente general de Integra S.A. por creer en el proceso de

Investigación, Desarrollo e Innovación I+D+i, con lo cual fue posible realizar esta investigación doctoral aplicada y por la confianza depositada en el equipo de trabajo.

Finalmente, agradezco a todos los integrantes de los grupos de investigación TransFórmate y DINOP por la contribución al desarrollo de esta investigación.

Rubén Iván Bolaños.

Resumen

La presente investigación está enfocada en la sofisticación de la planeación de los sistemas de transporte público de pasajeros operado con buses de tránsito ligero, denominados BRT, específicamente se analizan y se resuelven dos problemas propios de la planeación operativa, los cuales se conocen como Programación de Vehículos (Vehicle Scheduling Problem) y Programación de Conductores (Crew Scheduling Problem), ambos problemas son catalogados como problemas de difícil solución y se clasifican en la literatura especializada como problemas de Tipo NP- Hard.

La descripción y contextualización de los problemas a resolver se realiza desde la planeación general de los sistemas de transporte público, donde se describen cada una de las etapas, las cuales corresponden a la planeación estratégica, táctica y operacional; para esta última etapa, se presenta el estado del arte relacionado a los Problemas de VSP y CSP durante las últimas 3 décadas. Los estudios abordados involucran tanto casos genéricos propios de la literatura especializada así como casos reales de la operación de sistemas de transporte a nivel mundial.

A partir de la revisión de la literatura, se describen ambos problemas desde el punto de vista matemático de manera general y acorde a modelos existentes para cada uno de ellos, seguidamente, se realizan ajustes y modificaciones a dichos planteamientos, de tal manera que permita modelar cada uno de los problemas desde las particularidades del caso real de la operación del sistema de transporte público de pasajeros del Área Metropolitana Centro Occidente de Colombia, denominado Megabús.

La investigación plantea como método de solución una metodología híbrida en cascada con dos enfoques; heurístico y matemático para conformar una solución Mateheurística, en la primera parte, se aborda el problema de VSP cuyos resultados óptimos permiten dar inicio a la metodología de CSP, la cual combina los métodos de secuenciamiento de Prins y la aplicación modificada del conocido algoritmo de Kruskal.

El segundo enfoque plantea un problema de Generación de Columnas, para lo cual, se desarrolla en primera medida un Problema de Programación Entera Mixta MIP, que involucra restricciones de asignación única y continua de conductores, que permite llevar a cabo la generación de cada una de las columnas. La metodología continúa con la implementación de un método de Bin Packing el cual representa el problema auxiliar y un método de Set Partitioning que representa el problema maestro, con los cuales se abarcan las restricciones de capacidad, descanso y regulaciones laborales propias del problema tratado. Los resultados obtenidos permiten mejorar la solución actual del sistema de transporte público real para ambos problemas abordados.

Finalmente, con el fin de validar la metodología propuesta, desde el punto de vista de la literatura especializada, se realiza un estudio comparativo con instancias de la literatura y la instancia del caso real, para lo cual, se emplean cortes y generación de columnas especializados para problemas de camino más cortos con recursos restringidos (RCSP), contenidos en un solver especializado denominado VRPSolver, para lo cual, se traducen las instancias de VSP en un problema de ruteamiento de vehículos con ventanas de tiempo (VRPTW).

Los resultados obtenidos permiten validar el desarrollo de la metodología propuesta, y los aspectos de planeación operativa concernientes a la programación de vehículos y programación de conductores para el caso real fueron mejorados de forma sobresaliente y de manera relevante en el estado del arte.

Abstract

This research is focused on the sophistication of the planning of public passenger transport systems operated with Bus Rapid Transit BRT, specifically two problems of operational planning are analyzed and solved, which are known as Vehicle Scheduling Problem VSP and Driver Scheduling Problem , both problems are classified as difficult to solve and are classified in the specialized literature as NP-Hard problems.

The description and contextualization of the problems to be solved is carried out from the general planning of the public transport systems where each of the stages that involves strategic, tactical and operational planning are described; for the last one stage, the state of the art involving the development of VSP and DSP during the last 3 decades is presented. The studies addressed involve both generic cases of the specialized literature as well as real cases of the operation of transportation systems worldwide.

Based on the literature review, both problems are described from the mathematical point of view in a general way and according to existing models for each of them, then adjustments and modifications are made, in such a way that it allows modeling both problems from particularities of the real case of the public passenger transport system of the Central Western Metropolitan Area from Colombia, called Megabús.

We proposes a hybrid cascade methodology as solution method with two approaches; heuristic and mathematical solution which is known as Mateheuristic method. In the first part, the

VSP problem is approached whose optimal results allow to start the CSP methodology, which starts with the heuristic solution that combines the Prins sequencing methods and the modified application of the well-known Kruskal algorithm.

The second approach raises a Column Generation Problem, for which a Mixed Integer Programming MIP developed, which involves restrictions on the assignment of driver with which the Columns Generation is carried out. The methodology continues with the implementation of a Bin Packing method as designed as an auxiliary problem and a Set Partitioning problem as a Master problem, with which the capacity restrictions, rest and labor regulations typical of the problem treated are covered. The results obtained allow to improve the current solution of the real public transport system for both problems addressed.

Finally, in order to validate the proposed methodology, from the point of view of the specialized literature, a comparative study is carried out with instances of the specialized literature and the real case, for which, cuts and generation of specialized columns are used to Resource Restricted Shorter Path Problems (RCSP), contained in a specialized solver called VRPSolver, for which VSP instances are translated into a Vehicle Routing Problem with Time Windows (VRPTW).

The results obtained allow to validate the development of the proposed methodology, and the operational planning aspects concerning the programming of vehicles and programming of drivers for the real case were improved in an outstanding and relevant way in the state of the art.

Índice general

Índice general	12
Índice de figuras	15
Índice de tablas	1
1. Introducción	2
1.1. Descripción del problema	2
1.2. Objetivos	4
1.2.1. Objetivo general	4
1.2.2. Objetivos específicos	4
1.3. Estructura del documento	5
2. Marco referencial	7
2.1. Planeamiento del servicio de transporte público de pasajeros	7
2.1.1. Decisiones estratégicas	8
2.1.2. Decisiones tácticas	8
2.1.3. Decisiones operacionales	9

2.2. Antecedentes	15
3. Formulaciones matemáticas	27
3.1. Formulación del problema de programación de conductores	28
3.1.1. Notación matemática	29
3.1.2. Modelo matemático	30
3.2. Formulación del problema de asignación de vehículos con múltiples depósitos	31
3.2.1. Definición de variables de decisión	33
3.2.2. Formulación matemática	34
4. Metodología	36
4.1. Enfoque heurístico	39
4.1.1. División de tablas	41
4.1.2. Construcción de turnos de trabajo	47
4.1.3. Resultados enfoque heurístico	50
4.2. Enfoque matemático	53
4.2.1. Notación matemática	55
4.2.2. Estudio para validación de metodologías	66
4.2.3. Implementación del estudio comparativo	72
5. Análisis resultados	82
5.1. Enfoque heurístico	82
5.2. Enfoques matemáticos	85
5.2.1. Comparativo metodologías enfoque matemático	90

5.2.2. Marco general y caso real	95
6. Conclusiones, trabajos futuros y producción	98
6.1. Conclusiones	98
6.2. Trabajos futuros	102
6.3. Producción	103
A.	106
A.1. Óptimos <i>GCSP</i> (Beasley, 96)	106
Bibliografía	118

Índice de figuras

2.1. Planeación del sistema de transporte público de pasajeros. Tomado de (Ibarra-Rojas et al., 2015)	11
4.1. Metodología híbrida de solución problema de programación de vehículos MDVSP. Tomado de Marín-Moreno (2019).	38
4.2. Metodología híbrida de solución problema de programación de vehículos MDVSP. Fuente: Elaboración propia.	39
4.3. Metodología general de CSP para Integra S.A. Fuente: Elaboración propia.	41
4.4. Subgrafo según la secuencia. Fuente: Elaboración propia.	42
4.5. Subgrafo sólo con conexiones factibles entre viajes. Fuente: Elaboración propia.	43
4.6. Digrafo para encontrar una partición balanceada de una ruta. Fuente: Elaboración propia.	44
4.7. Ruta más corta que corresponde a una partición balanceada de la tabla de servicios. Fuente: Elaboración propia.	46
4.8. Agrupación de viajes en bloques de trabajo. Fuente: Elaboración propia.	46
4.9. Grafo ilustrativo Algoritmo Kruskal. Fuente: Elaboración propia.	47

4.10. Árbol de expansión mínima. Fuente: Elaboración propia.	48
4.11. Kruskal modificado (camino mínimos). Fuente: Elaboración propia.	49
4.12. Kruskal modificado construcción turnos. Fuente: Elaboración propia.	50
4.13. Diagrama de flujo algoritmo solución. Fuente: Elaboración propia.	54
4.14. Componentes del estudio comparativo. Fuente: Elaboración propia.	80
5.1. Resultado turnos de trabajo caso Megabús 2019: Amarillo Alimentación, Verde Articulados. Fuente: Elaboración propia.	84
5.2. Tiempo de cómputo librería CSP genérico completa, con número fijo de conductores. Fuente: Elaboración propia.	91
5.3. Comparativo minimización de número de conductores. Fuente: Elaboración propia.	92
5.4. Comparativo metodologías que alcanzan BKS para número de conductores. Fuente: Elaboración propia.	94
5.5. Metodología Integrada - Sistema de Transporte Masivo. Fuente: Elaboración propia.	96
5.6. Efecto de la primera fase de la metodología. Tomado de: Marín-Moreno (2019). . .	96
A.1. BKS itinerarios <i>csp50</i>	108
A.2. BKS itinerarios <i>csp100</i>	109
A.3. BKS itinerarios <i>csp150</i>	110
A.4. BKS itinerarios <i>csp200</i>	111
A.5. BKS itinerarios <i>csp250</i>	112
A.6. BKS itinerarios <i>csp300</i>	113
A.7. BKS itinerarios <i>csp350</i>	114

A.8. BKS itinerarios <i>csp400</i>	115
A.9. BKS itinerarios <i>csp450</i>	116
A.10. BKS itinerarios <i>csp500</i>	117

Índice de tablas

4.1. Comportamiento de la metodología en los casos de operación de Integra S.A.	52
4.2. Resultados obtenidos por el coordinador y el modelo propuesto para la instancia Mega060519	66
5.1. Comparativo metodologías con número de conductores Fijo K	88
5.2. Comparativo metodologías minimizando el número de conductores K . Los GAPS son relativos Beasley96 y Derigs2014	89

Capítulo 1

Introducción

1.1. Descripción del problema

Actualmente alrededor del 55% de las personas a nivel mundial viven en zonas urbanas (Mundial, 2015), bajo un contexto de economía globalizada que, sin lugar a dudas, ha generado cambios significativos en los hábitos y costumbres de la población. Si bien, dicho fenómeno ha dinamizado la economía en los diferentes países, también es cierto que las ciudades han experimentado diversos problemas relacionados con la sobrepoblación, lo que ha conllevado un crecimiento de las mismas, de manera natural, hacia su periferia, obteniendo como resultado una expansión desordenada y un crecimiento desproporcionado de la ciudad, donde su principal problema es la movilidad de sus habitantes.

Ante este escenario, el transporte público de pasajeros toma gran importancia en las funciones básicas de las áreas urbanizadas y determina la forma de vida de la ciudades, siendo el confort en la movilidad una de las principales medidas de calidad de vida de las personas. En este sentido, el uso del transporte público de pasajeros tiene un papel fundamental a la hora de planear el territorio

en su conjunto, esencialmente desde el punto de vista social, económico y ambiental.

El crecimiento acelerado de algunos sectores de la economía ha determinado las costumbres diarias de viaje de los ciudadanos, llevando a una expansión desordenada de las ciudades, definiendo lugares de trabajo, estudio y vivienda sin una planeación de las mismas, generando mayor congestión del tráfico, desigualdad y segregación social; estas circunstancias se agravan por la necesidad de disminuir las consecuencias del cambio climático, la contaminación del aire y las muertes en accidentes de tránsito, adicionalmente, en los últimos años se suma la disminución del uso de transporte público de pasajeros debido a la adquisición de medios de transporte particulares.

Ante esta creciente problemática los gobiernos locales y nacionales a nivel mundial han venido realizando ingentes esfuerzos con el fin de consolidar un sistema de transporte público robusto, eficaz y con estándares elevados de confort. Una de las mayores apuestas a nivel mundial en cuanto a la reestructuración del transporte público de pasajeros, se consolida en los sistemas masivos e integrados de transporte público, especialmente aquellos basados en tránsito ligero, como son los sistemas de metro y buses de tránsito ligero (BRT por sus siglas en inglés de *Bus Rapid Transit*), siendo este último, el sistema de mayor implementación en los países de Latinoamérica; movilizan diariamente alrededor de 21 millones de pasajeros por día, lo cual representa el 62 % de los pasajeros diarios movilizadas en sistemas BRT a nivel mundial ([BRTData, 2020](#)). Estos sistemas son atractivos debido a que su costo de infraestructura es bajo comparado con el metro, minimiza los tiempos de viaje al contar con carriles exclusivos, estaciones de paradas fijas y sistema de pago sin usar dinero en efectivo.

Aún así, los problemas de movilidad seguirán aumentando, ya que de acuerdo a cifras de la ONU ([Mundial, 2015](#)), se calcula que un 90% de la expansión urbana en los próximos decenios,

se producirá en países que se encuentran en vía de desarrollo, tal como es el caso de los países latinoamericanos, por ende, desde hoy las ciudades están obligadas a pensar en planes para mitigar dichos problemas, de tal manera que sea posible desplazar personas de manera rápida, eficaz y económica.

Dicha necesidad no es ajena a las políticas mundiales actuales, tal como se refleja a través de la Organización de las Naciones Unidas ONU, en su política de Objetivos de Desarrollo Sostenible ODS (PNUD, 2020), donde se establece como una de las metas del objetivo once (11) para el año 2030, proporcionar acceso a sistemas de transporte seguros, asequibles, accesibles y sostenibles para todos y mejorar la seguridad vial, en particular mediante la ampliación del transporte público.

Ante este panorama, es indispensable el desarrollo de investigaciones académicas enfocadas a la sofisticación de la planeación de los sistemas de transporte público en cada una de sus etapas.

1.2. Objetivos

1.2.1. Objetivo general

Solucionar de manera conjunta el problema de programación de vehículos y conductores de un sistema de transporte público de pasajeros con aplicación real.

1.2.2. Objetivos específicos

- Realizar un estudio del estado del arte del problema de programación de vehículos y del problema de programación de conductores.
- Definir un modelo matemático que involucre los problemas de programación de vehículos y

conductores.

- Ajustar el modelo matemático a las necesidades y restricciones prácticas del caso real de operación de la empresa Integra S.A.
- Realizar una revisión de las técnicas metaheurísticas y exactas que puedan ser aplicadas en la solución del modelo matemático ajustado.
- Diseñar e implementar algoritmos matemáticos eficientes aplicados al problema de programación de vehículos y conductores.
- Validar la metodología propuesta usando un software de optimización comercial en casos de prueba de la literatura especializada y en el caso real de la empresa Integra S.A.

1.3. Estructura del documento

El estudio realizado, después del presente capítulo, donde se especifican los objetivos y el alcance de la investigación, se encuentra desarrollado de la siguiente manera: Inicialmente, en el capítulo 2, se presenta la estructura sugerida por la literatura especializada para la toma de decisiones al constituir y operar servicios para transporte público de pasajeros. Luego, en el capítulo 3, se formalizan los componentes de la planeación operativa contextualizada en 2 a través de modelos matemáticos que servirán de punto de partida de las propuestas metodológicas. Posteriormente, en el capítulo 4 se presenta la metodología integrada de dos fases, enfatizando en que este estudio profundiza el aspecto concerniente a la programación de conductores, al contar previo al estudio con una metodología que hace parte del estado del arte de la programación de vehículos y ajustada al sistema de transporte público masivo estudiado en este trabajo.

Consolidando los buenos resultados obtenidos en los experimentos realizados, en el capítulo 5 se presenta el rendimiento de las metodologías propuestas, mostrando sus aportes en diferentes contextos. En el capítulo 6 se presentan las conclusiones que arroja el estudio, los trabajos futuros que se recomiendan después de la experiencia adquirida, y un recuento de la producción académica generada durante el transcurso de este proyecto de investigación aplicada.

De manera complementaria se presenta un apéndice con la especificación de las mejores soluciones conocidas BKS (Del inglés: *Best Known Solutions*) aportadas en este trabajo para el problema genérico de programación de conductores (Del inglés: *Generic Crew Scheduling Problem*), que pueden ser del interés para futuras investigaciones, que como esta, priorizan la minimización de flote y de personal.

Capítulo 2

Marco referencial

2.1. Planeamiento del servicio de transporte público de pasajeros

Un diseño eficiente del sistema de transporte público conlleva a grandes beneficios, entre los que se destacan, la satisfacción de los pasajeros y la reducción considerable de aspectos como la contaminación del medio ambiente, la congestión del tráfico, los tiempos de viaje y los costos operacionales ([Ibarra-Rojas et al., 2015](#)).

La principal dificultad que se debe enfrentar a la hora de realizar las tareas de planeación y operación del transporte público de pasajeros, se da en la búsqueda de un equilibrio entre el nivel de satisfacción percibido por los usuarios en la prestación del servicio y los costos asociados a las operación misma, de esta manera, los usuarios buscan que los recorridos de cada viaje se realicen de manera rápida, los tiempos de espera y el número de transbordos sean mínimos, así como estándares mínimos de confiabilidad y confort.

Por otra parte, las empresas operadoras del sistema de transporte público buscan maximizar la rentabilidad y minimizar la cantidad de vehículos y conductores necesarios para la operación. El proceso de planificación se extiende por todas las decisiones que se deben tomar antes de la operación del sistema, dicho proceso se denomina en la literatura especializada como problema de Planeamiento de la Red de Transporte (TNP del inglés *Transit Network Planning problem*); el problema visto desde una perspectiva general se torna complejo y de difícil solución, razón por la cual, se divide en tres fases que abarcan las decisiones tácticas, estratégicas y operacionales (Ceder, 2007).

Por lo tanto, el problema es solucionado en cascada, es decir, cada una de las etapas se realiza de manera secuencial y la salida de una etapa representa la entrada de la siguiente; para ello se describen los siguientes procesos.

2.1.1. Decisiones estratégicas

- **Diseño de la Red de Transporte (TND del inglés *Transit Network Design*):** En esta etapa se realizan los diseños de las líneas y las características operativas asociadas, tal como el espacio entre los paraderos, tamaño de las estaciones y rutas. Esta etapa busca la minimización de la suma ponderada de los costos asociados a los operadores y los usuarios.

2.1.2. Decisiones tácticas

- **Conjunto de Frecuencias (FS del inglés *Frequency Setting*):** Caracteriza los periodos de la operación basados en los patrones de demanda como horas pico, horas valle y determina la cantidad de viajes necesarios para cubrir toda la demanda en cada periodo planeado.

- **Horario de la Red de Transporte (TNT del inglés *Transit Network Timetabling*):** Define los tiempos de llegada y salida de cada uno de los buses en todas las paradas de la red de transporte, de tal manera que se cumpla con la frecuencia definida, la demanda patrón específica y se minimicen los tiempos de espera.

2.1.3. Decisiones operacionales

- **Problema de Programación de Vehículos (VSP del inglés *Vehicle Scheduling Problem*):**
Asigna cada viaje planeado a los vehículos disponibles de tal forma que los costos asociados al uso de los vehículos sea minimizado ya sea con viajes en vacío o con demanda de pasajeros.
- **Problema de Programación de Conductores (CSP del inglés *Crew scheduling Problem*):**
Involucra las tareas diarias que cubren todos los viajes programados y minimiza la nómina de los conductores. Una solución del CSP debe satisfacer las regulaciones laborales tales como mínimo y máximo tiempo de trabajo diario, máximo tiempo continuo de trabajo sin descanso y el tiempo de descanso diario requerido para todos los conductores.
- **Problema de Distribución de Conductores (DRP del inglés *Driver Rostering Problema*):**
Dado un conjunto genérico de tareas en un horizonte de tiempo determinado (semana, mes o año) para los conductores asignados a un depósito en particular, el DRP asigna ese conjunto de tareas a los conductores disponibles para programar su trabajo, lo cual se denomina lista o tabla de distribución (Roster), de tal manera que se cumplan con las regulaciones laborales y se busque un bienestar laboral para los conductores.

La solución de cada una de las fases descritas anteriormente determina el proceso de Planeamiento

de la Red de Transporte. Sin embargo, la operación del sistema de transporte es afectada por la incertidumbre en los tiempos de viaje, patrones de demanda y perturbaciones tales como actividades de protesta, daño de los vehículos que se encuentran en operación, condiciones climáticas y actividades del día a día, condiciones que no son tenidas en cuenta al momento de realizar el Planeamiento del Sistema de Transporte, siendo esta la razón de adoptar una estrategia de control en tiempo real, que busque garantizar la eficiencia en el servicio durante la operación del sistema.

Estrategias adoptadas en el control de la operación

- **Control en Estaciones:** Permite tomar decisiones tales como retener un determinado vehículo en algunas de las paradas o saltar una de ellas con el fin de aumentar la frecuencia del servicio, reducir el tiempo de viaje o de espera, y permitir la transferencia de pasajeros.
- **Control entre estaciones:** Algunas de las estrategias adoptadas en esta etapa son control de la velocidad de operación y prioridad en las señales de tránsito.

Cabe resaltar que, el control en tiempo real es fruto del beneficio que brinda el avance tecnológico con el que se cuenta hoy en día; como son las herramientas de sistemas automáticos de recaudo (AFC), sistemas automáticos para el conteo de pasajeros (APC), sistemas automáticos de localización de vehículos y sistemas de posicionamiento global (GPS) (Ibarra-Rojas et al., 2015).

La figura 2.1 presenta la manera como interactúa cada una de las etapas del proceso de planeación del sistema de transporte público de pasajeros.

Cada uno de los problemas asociados a cada etapa de la planeación del transporte público de pasajeros, ha sido ampliamente estudiado en la literatura especializada, y debido su complejidad

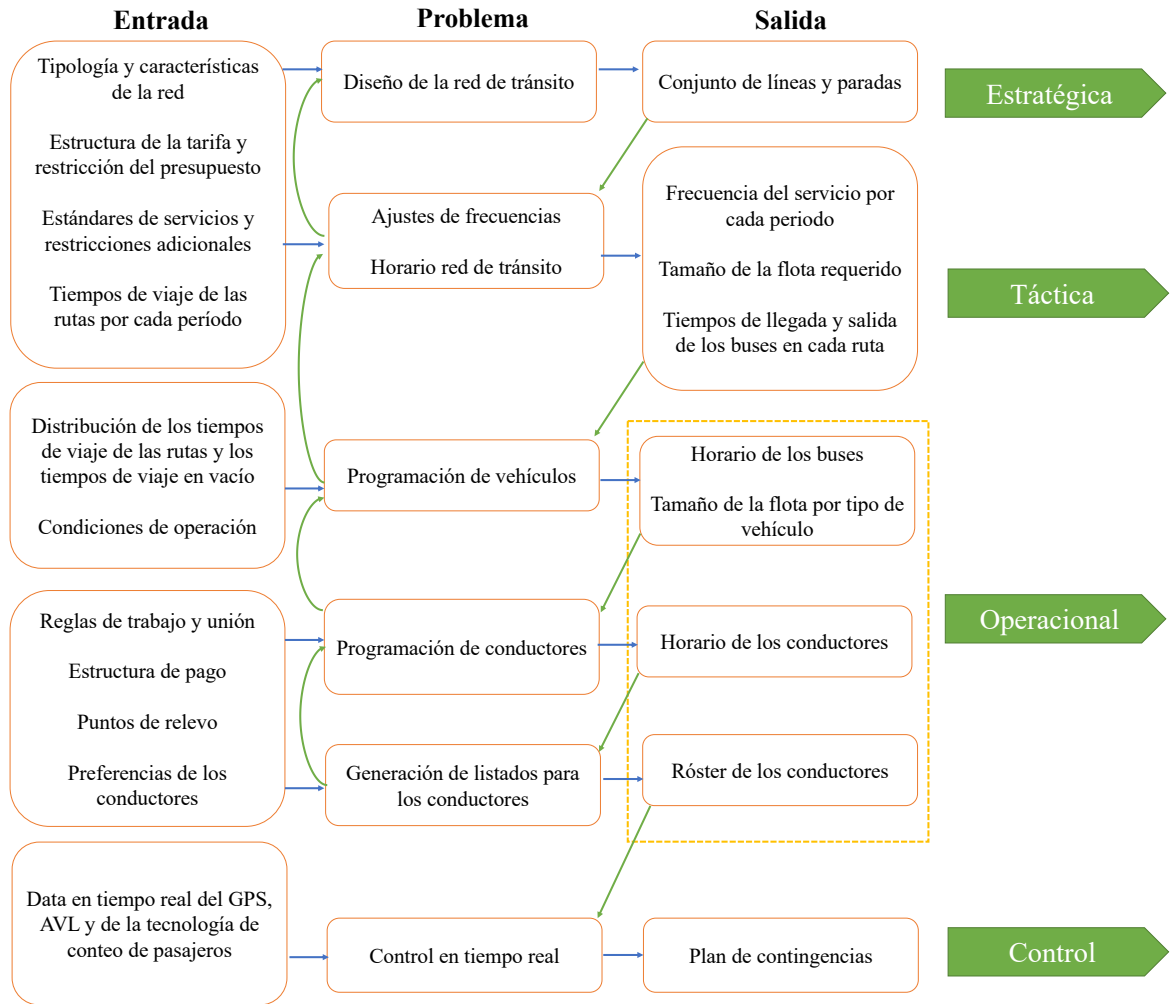


Figura 2.1: Planeación del sistema de transporte público de pasajeros. Tomado de (Ibarra-Rojas et al., 2015)

matemática y computacional se catalogan como problemas de tipo NP-Hard, es decir, que no existe un método o algoritmo que sea capaz de resolver el problema de manera general en tiempos de cómputo razonables. Así las cosas, cada subproblema se resuelve de manera separada, es decir, el problema general de planeación se soluciona de manera secuencial.

Partiendo de la complejidad del problema, en esta investigación se busca resolver la etapa correspondiente a la Planeación Operacional, específicamente se propone resolver los problemas asociados a la Programación de Vehículos VSP y Programación de Conductores CSP. Solucionar

estos dos problemas se sustenta en que los principales costos que asumen las compañías de transporte público de pasajeros, son los costos operativos relacionados a la compra y uso de la flota y la nómina de sus conductores, razón por la cual, es estratégicamente recomendado, contar con herramientas computacionales basadas en algoritmos matemáticos de optimización que ayuden no sólo a realizar la planeación operativa, sino que, ayuden a reducir los costos asociados.

El objetivo de los algoritmos computacionales es determinar cada turno de trabajo o itinerario, compuesto por diferentes viajes y a su vez, asignar la totalidad de los viajes a los vehículos disponibles.

Tradicionalmente, dichos problemas presentan una estructura secuencial en su solución, es decir, se soluciona primero un problema para que luego con los resultados obtenidos se haga la optimización del segundo, lo que se conoce como una optimización paso a paso.

De manera más detallada, el primero de los subproblemas se conoce como Programación de Vehículos con Múltiples Depósitos MDVSP (del inglés *Multiple Depot Vehicle Scheduling Problem*), en el cual se determina un conjunto de vehículos que debe realizar un conjunto de viajes de un conjunto de rutas con una frecuencia dada en cada momento del día. El segundo de los subproblemas se denomina Programación de Conductores CSP del inglés *Crew Scheduling Problem*, en el cual, se determinan los turnos de trabajo que deben ser asignado a un conductor, los cuales están sujetos a restricciones laborales y operacionales.

Si bien, en muchos casos la solución hallada de manera separada en cada problema, permite reducir de manera significativa tanto el tiempo computacional como los costos asociados a cada subproblema, no es posible determinar si dichas soluciones son óptimas globales, dado que, no se tiene en cuenta la integralidad de los problemas, razón por la cual, uno de los enfoques que se

aborda en el método de solución propuesto en esta investigación, es integrar ambos problemas en un único modelo matemático, el cual será validado en casos de prueba de la literatura especializada y en un caso de aplicación real, esta última obedece a que la presente propuesta se desarrolla en el sector empresarial bajo la modalidad de formación doctoral en la empresa.

En una revisión preliminar del estado del arte de la programación de vehículos y conductores, se identificó que existen diferentes investigaciones aplicadas al mejoramiento de la operación del sistema de transporte público de pasajeros mediante la optimización del Problema MDVSP y CSP, las cuales han sido de suficiente importancia para las empresas que han implementado sus resultados, ya que le aportan al desarrollo de un sistema de transporte eficiente, capaz de atender las necesidades de movilidad requeridas en las urbes ([Ibarra-Rojas et al., 2015](#); [Muñoz and Paget-Seekins, 2016](#)).

La creación de itinerarios es una de las tareas más complejas en la planeación operativa de los sistemas de transporte público de pasajeros, ya que requiere gran cantidad de tiempo en la gestión de la operación, siendo este a su vez, un problema de gran interés dado que, los costos asociados son los más elevados dentro del presupuesto de la operación de los sistemas de transporte público de pasajeros ([Ceder, 2007](#)). Sin embargo, independiente de las variables presentes en cada subproblema, el objetivo siempre estará enmarcado en el cumplimiento total de los itinerarios y en la reducción de costos relacionados con la operación del sistema a través de procesos de optimización.

Cada itinerario es una descripción de los viajes que se deben ejecutar en tiempos y tramos específicos denominados rutas, obedeciendo a una frecuencia según las condiciones del servicio y las necesidades del servicio público del transporte determinadas por la planeación táctica; es decir,

la combinación de ruta y tiempo de salida se denomina servicio y una agrupación de servicios del mismo tramo es definida como tabla o turno; las rutas de los sistemas de transporte público se definen desde su planeación estratégica y por lo general no poseen cambios sustanciales ni en el corto ni el mediano plazo.

Cada ruta debe ser servida con una frecuencia determinada, a una velocidad promedio definida, esto debe definirse en la planeación táctica del sistemas de transporte y todos estos requerimientos de las rutas son definidos desde el diseño mismo de la red de servicio buscando satisfacer las necesidades identificadas en la planeación estratégica.

Ante este escenario real de la problemática asociada a la operación de los sistemas de transporte público de pasajeros, se plantea en esta investigación la pregunta ¿Cuál es la metodología de optimización que permita solucionar de manera conjunta el problema de la operación del sistema de transporte público de pasajeros utilizando métodos mateheurísticos de optimización.

En esta investigación, se propone solucionar un problema de la Planeación Operativa de un Sistema Transporte Público de Pasajeros a través del desarrollo e implementación de una metodología que combine algoritmos heurísticos y mateheurísticos capaces de resolverlo en tiempos de cómputo razonables.

Para lo cual, en una primera fase, se resolverá el problema de programación de conductores, para ello, se realizará la revisión de los modelos matemáticos que describen este problema, a partir de los cuales se propondrá un nuevo modelo matemático que sea ajustado al caso real de una empresa del sector transporte público, así mismo, cada uno de los algoritmos y modelos de solución aquí propuestos serán validados con casos de la literatura especializada y casos reales mediante la comparación numérica de las soluciones obtenidas.

En una segunda fase se resolverá el problema de programación de vehículos con múltiples depósitos, de igual manera serán validados los modelos en casos de la literatura y finalmente serán aplicados al caso real, es preciso resaltar que los métodos publicados en la literatura especializada con aplicaciones a casos de prueba involucran hasta 500 viajes diarios y cuatro depósitos.

El problema a solucionar en esta investigación, presenta mayor complejidad por tratarse de dos problemas de tipo combinatorio con una gran cantidad de variables, además, cada uno de ellos considera restricciones técnicas y regulatorias de la operación real, tal como, máximo tiempo continuo de trabajo, tiempo total de trabajo, tiempos de descanso, viajes en vacío entre otras. También es preciso resaltar que, se diseñará un nuevo caso de prueba con datos reales de la programación de viajes de transporte masivo de la empresa Integra S.A., que quedará disponible y podrá ser utilizado en futuras investigaciones.

Finalmente, en la tercera fase se propone el diseño y desarrollo de una metodología de solución que describa de manera conjunta los problemas de MDVSP y CSP, el cual será solucionado mediante una técnica de solución mateheurística cuya aplicación sera realizada directamente en el caso real, toda vez que, la validación de cada método se realizará en los problemas separados en las fases 1 y 2.

2.2. Antecedentes

Los problemas relacionados a las operaciones de transporte han sido altamente estudiados, alcanzando soluciones eficientes para múltiples aplicaciones reales. Estos se pueden clasificar como problemas de enrutamiento de vehículos o problemas de agendamiento. Para ambos, las metodologías de solución propuestas se basan en enfoques exactos y aproximados, intentando

considerar la mayor cantidad de características que representen un sistema real, de tal manera que se generen buenas soluciones inclusive en instancias de gran porte y casos singulares del problema.

El estudio de los problemas asociados a los sistemas de transporte público de pasajeros tipo BRT tienen gran importancia y peso sobre todo en las compañías encargadas de operar dichos sistemas. Las operaciones de un sistema de transporte masivo en autobuses suelen incluir unos costos elevados, debido a que requiere la compra y mantenimiento de vehículos, y contratación y administración de personal especializado (como conductores con licencias de conducción para autobuses articulados).

A partir de los servicios demandados, en la fase operativa se debe resolver la programación del conjunto de vehículos disponibles para el desarrollo de cada Timetable, generalmente, dicha flota se encuentra dispuesta en diferentes lugares geográficos denominados depósitos o patios de mantenimiento, por lo que esta fase, se representa mediante un problema de programación de vehículos con múltiples depósitos (MDVSP), donde se busca establecer la cantidad mínima de vehículos necesarios para llevar a cabo los servicios demandados, factor crítico debido a los altos costos que representa la compra y mantenimiento de un autobús. Una revisión de este problema se presenta en [Desaulniers and Hickman \(2007\)](#).

Posteriormente, para cumplimiento de la operación se deben desarrollar los turnos de trabajo en donde se vincule el horario del trabajo del conductor y el conjunto de servicios a realizar (recorrido y vehículo asignado), esta fase es conocida en la literatura como problema de la programación de turnos de trabajo, en la cual se busca generar la menor la cantidad de turnos necesarios para cumplir con los servicios demandados, debido a que cada turno implica costos de nómina y administración de la contratación (dotaciones, capacitaciones, departamento de recursos

humanos, entre otros). Una clasificación y revisión de este problema es presentada por [Van den Bergh et al. \(2013\)](#).

La etapa posterior a la programación de turnos, consiste en generar una distribución de turnos o agendamiento de los conductores de la nómina que permita cumplir los requerimientos operativos (tipos de licencia de conducción, horarios de estudio, manejo de segundo idioma, entre otros), buscando balancear la carga laboral y mejorar el bienestar laboral. Este problema en la literatura se conoce como Staff scheduling and rostering. Una revisión extensa de este se presenta en [Ernst et al. \(2004\)](#), allí se muestran las características principales en diferentes aplicaciones y la importancia de llegar a resolver este problema.

Por último, el control diario de la operación conlleva resolver problemas relacionados con fallas, que puedan afectar las programaciones y asignaciones ya establecidas, tanto de conductores como vehículos. Lo que se traduce en establecer un plan de recuperación; [Yu and Qi \(2004\)](#) presentan una revisión de los distintos esquemas de trabajo, siendo la formulación de planes (programaciones) robustos la más acogida en la práctica.

Para el caso del CSP, uno de los primeros trabajos con aplicación real, es el método propuesto por [Smith and Wren \(1988\)](#), en el cual presentan una formulación basada en el problema de cobertura (Set Covering Problem SCP), para lo cual, los autores proponen en primera medida la generación de conjuntos de turnos de manera exhaustiva bajo restricciones de un programador experto; seguidamente aplican dos estrategias de reducción de turnos. A partir de la reducción de conjuntos se resuelve un problema de SCP relajado, es decir, se resuelve un problema de programación lineal considerando las variables de decisión continuas, con el cual se obtiene una solución entera a través el método de Branch and Bound.

[Desrochers and Soumis \(1989\)](#) presentan un método de Generación de Columnas (CG), el cual se descompone en dos partes; primero se usa un problema de cobertura para elegir una solución factible de un día de trabajo; y seguidamente se resuelve un problema de camino más corto con restricciones de recurso, con lo cual se propone una nueva programación de trabajo diario que mejora la solución obtenida resolviendo el problema de cobertura. Los resultados obtenidos sobre instancias reales demuestran el buen desarrollo de la metodología propuesta.

Una de las formulaciones clásicas en la literatura para resolver el CSP es el enfoque de Partición de Conjuntos (SP) con restricciones de lugar, la cual se propone en el trabajo realizado por [Mingozzi et al. \(1999\)](#). En la metodología propuesta, cada columna de la matriz del SP corresponde a un turno de trabajo factible, el cual está compuesto por un conjunto de servicios que deben ser realizados por un operador. Adicionalmente, los autores describen un proceso para determinar el límite inferior de la solución óptima por medio de una solución heurística para el problema dual de la relajación lineal del SP, la solución del problema dual es utilizada para reducir el número de variables en el SP y de esta manera resolver el problema de resultante de SP a través de un algoritmo de Branch-and-Bound. La metodología es analizada en instancias de la literatura con un tamaño que varía entre 50 y 500 viajes.

Desde el punto de vista matemático, el CSP encierra más de un objetivo que se encuentran en conflicto, como lo son los costos y la calidad del servicio, este problema es abordado por [Lourenço et al. \(2001\)](#), para el cual desarrollan un algoritmo genético y una búsqueda tabú. Los algoritmos meta-heurísticos propuestos presentan características innovadoras que guían el proceso de búsqueda de manera eficiente y encuentran soluciones de buena calidad, además, las características propias de dichos métodos pueden ser aplicadas al desarrollo de heurísticas para

diferentes problemas de optimización de tipo combinatorio.

De manera similar, [Li and Kwan \(2003\)](#) presentan un algoritmo genético de dos objetivos, en el cual se usa un algoritmo heurístico goloso para construir la programación de turnos de trabajo por medio de la selección de cambios secuenciales en posibles itinerarios, para la evaluación de dichos cambios, los autores utilizan un método de lógica difusa, cuyo criterio está basado en el conocimiento de expertos de empresas de transporte público de pasajeros. Los resultados obtenidos en casos reales son prometedores, comparadas con las programaciones manuales que realiza cada una de las empresas involucradas.

En [Dias et al. \(2002\)](#) desarrollan un algoritmo genético que incorpora conocimiento de expertos en programación de turnos de trabajo para empresas de transporte público de pasajeros; el enfoque natural inmerso en dicho tipo de algoritmos, permite obtener soluciones que pueden ser implementadas directamente en el proceso de operación diaria. El algoritmo es implementado en casos reales de asignación de tripulación en compañías aéreas y en empresas operadoras de sistemas de transporte público de pasajeros.

Las empresas operadoras de transporte público también han despertado gran interés por los distintos tipos de Software que les permita realizar parte de su operación diaria de manera automática, como es el caso del trabajo publicado por [Fores et al. \(2002\)](#), quienes a través de un Software denominado TRACS II logran implementar un problema combinado de programación entera basada en una solución heurística, dicha solución es aplicada en diferentes escenarios de problemas de planeación y operación de empresas de transporte público de pasajeros y empresas de trenes en el Reino Unido.

Cada vez más, los Software empleados en la operación del transporte, deben ser ajustados a

las realidades propias y suplir las necesidades operativas del día a día, siendo así, el problema de programación de conductores con múltiples depósitos MD-CSP unas de las variables que busca aproximarse a las realidades presentes en muchas empresas dedicadas a la operación del transporte; dicha variante está representada en el trabajo realizado por [Boschetti et al. \(2004\)](#), quienes introducen un nuevo procedimiento para limitar la búsqueda en el espacio de soluciones basados en los métodos de Relajación de Lagrange y Generación de Columnas con restricciones del bien conocido Problema de Programación de Vehículos VSP; la metodología es probada en bases de datos generadas aleatoriamente y en casos reales.

[Abbink et al. \(2005\)](#), presentan un nuevo modelo para resolver problemas de programación de más de 6500 conductores aplicado en la empresa de trenes más grande Holanda. La metodología propuesta permite el desarrollo de un modelo de producción denominado “*Sharing Sweet Sour*”, que satisface la programación de conductores e incrementa la puntualidad y la eficiencia de los servicios planeados para cada día de trabajo.

[Zhao \(2006\)](#), desarrolla un enfoque heurístico denominado “ZEST for ESTimator” con el cual se genera una estimación del número de conductores requeridos para una programación de viajes dada. La estimación realizada, se basa en la observación de los conductores necesarios en los periodos de máxima demanda en los turnos de la mañana, tarde y noche, para cual se resuelve cada sub-problemas acorde a la franja horaria y finalmente combina las soluciones encontradas; finalmente, el éxito del método radica en la incorporación de procedimientos que se realizan de manera manual por expertos.

[Yen and Birge \(2006\)](#) describen un modelo estocástico de programación entera para resolver el problema de asignación de tripulación en líneas aéreas, para lo cual desarrollan un algoritmo de

ramificación que permite identificar conexiones costosas entre los vuelos a realizar. El algoritmo de ramificación utiliza una técnica que le permite ramificar variables de manera simultánea sin eliminar la optimalidad del espacio de solución.

[Takahashi et al. \(2008\)](#) proponen un algoritmo para modificar de manera automática la programación de conductores cuándo suceden imprevistos en la operación diaria; el método inicia con una solución relajada del problema y en una segunda etapa se aplica un algoritmo de Búsqueda Tabú para minimizar la cantidad de restricciones que fueron violadas en la primera etapa, de esta manera, el algoritmo es capaz de encontrar soluciones que satisfacen todas las restricciones del problema y adicionalmente la solución encontrada es por lo menos de igual calidad que la solución hallada de manera manual.

[Steinzen et al. \(2009\)](#), proponen un método de solución que mejora de manera regular la solución mientras integra asignación de vehículos y de conductores; los autores proponen un método de dos fases, en la primera parte solucionan un programa lineal relajado de la formulación de recubrimiento de conjuntos usando el método de Generación de Columnas junto con una técnica de relajación de Lagrange. En la segunda fase, la metodología genera soluciones enteras combinando diferentes versiones del algoritmo de ramificación; los resultados numéricos obtenidos de casos reales y de prueba muestran mejoras significativas en la programación de la tripulación en cada caso.

[Portugal et al. \(2009\)](#), proponen un nuevo modelo para formular el problema de programación de conductores, el cual se basa en los modelos de recubrimiento y partición de conjuntos, cuyo éxito radica en que en el modelo propuesto se tienen en cuenta tanto los objetivos de las empresas operadoras como las necesidades de confort de los usuarios.

[Chen and Niu \(2012\)](#), resuelven el problema de programación de conductores considerando tres periodos de operación en el día; mañana, tarde y noche, incluyendo restricciones de cambio de turnos e intensidad de trabajo. Para solucionar el problema, los autores proponen un modelo de optimización formulado como un problema de programación entera binaria y como técnica de solución se implementa un algoritmo de Búsqueda Tabú. La metodología es analizada de manera completa en un caso de estudio de la literatura especializada.

[Azadeh et al. \(2013\)](#), implementan un algoritmo de Optimización Cúmulo de Partículas (PSO) sincronizado con una heurística de Búsqueda Local para solucionar el problema de asignación de tripulación de aerolíneas, en el cual se incluye como objetivo, minimizar los costos de operación teniendo en cuenta restricciones legales y regulaciones impuestas. El método es comparado con soluciones encontradas por medio de un Algoritmo de Colonia de Hormigas y un Algoritmos Genético, con lo cual se demuestra la eficiencia y superioridad del algoritmo PSO híbrido propuesto.

[Tóth and Krész \(2013\)](#), presentan una metodología basada en cortes y uniones que permite generar soluciones iniciales de manera rápida para el CSP; debido a la flexibilidad del enfoque propuesto, es posible desarrollar un método heurístico iterativo con lo que se obtiene un algoritmo más robusto para solucionar el problema de la asignación de conductores en instancias reales.

[Chen and Shen \(2013\)](#), desarrollan una nueva estrategia de Generación de Columnas para mejorar la eficiencia de los enfoques de programación de tripulaciones, propuestas en casos reales, para lo cual, los autores construyen un conjunto de “buenos turnos” a partir de la solución de un problema de Conocimiento Específico y a partir de dichas soluciones preseleccionadas, se resuelve el problema general mediante un algoritmo de CG. Los experimentos realizados sobre un conjunto

de casos de problemas reales en China muestran que el proceso de solución es significativamente mejorado en cuanto al tiempo de ejecución de cada problema.

[Kecskeméti and Bilics \(2013\)](#), solucionan el problema de programación de conductores usando como método de solución un algoritmo híbrido evolutivo y un problema de programación entera, en la primera parte, los autores usan un enfoque del problema de partición de conjuntos, solucionando el modelo con un algoritmo de CG en un esquema de Branch-and-Bound. Con el fin de aplicar la metodología en problemas reales de gran tamaño, los autores combinan el método de programación entera con un algoritmo evolutivo goloso de estado estacionario. Los resultados obtenidos en problemas de programación de conductores son cercanos a las soluciones óptimas del problema y fueron obtenidos en un tiempo de cómputo razonable.

[Shen et al. \(2013\)](#), presentan un enfoque evolutivo de adaptación que incorpora un Algoritmo Genético híbrido para resolver problemas de asignación de tripulación en sistemas de transporte público, para asegurar la eficiencia del algoritmo, la codificación del problema se realiza sobre un cromosoma cuyo tamaño puede variar durante el proceso iterativo. Durante el proceso del GA, la adaptación del tamaño de los cromosomas se logra mediante operaciones genéticas de cross-over y mutación con estrategias de remoción y reposición asistidas por un algoritmo goloso simple. La metodología es validada en once instancias reales del problema de programación de conductores en China, encontrando mejores soluciones en cada uno de los casos con respecto a las soluciones existentes.

[Tian and Songa \(2013\)](#), proponen un algoritmo de Colonia de Hormigas mejorado para resolver el problema de asignación de tripulación en líneas de trenes ligeros en China, los autores dividen el plan de trabajo en dos fases, enfocándose en la primera fase que consiste en determinar

los costos de cada plan. Los experimentos realizados con datos de prueba de sistemas de trenes de alta velocidad de la ciudad de Beijing muestran el buen desarrollo del algoritmo.

[Hanafi and Kozan \(2014\)](#), diseñan un modelo matemático para la programación de la tripulación de trenes con el objetivo de minimizar el número de tareas de cada uno de los conductores, para lo cual se reducen los tiempos de ocio en cada transición; el modelo involucra un número elevado de variables de decisión y restricciones, razón por la cual se propone como método de solución un algoritmo heurístico constructivo híbrido con una búsqueda de Recocido Simulado (Simulated Annealing), la metodología permite encontrar soluciones cercanas a las óptimas en tiempos de cómputo razonables para problemas de gran tamaño.

Los autores de [Fuentes et al. \(2015\)](#) proponen un nuevo modelo para solucionar el problema de programación de la tripulación en redes de tránsito ligero, para solucionar el modelo, los autores solucionan el problema de relajación de Lagrange aprovechando la descomposición de nominada “ad hoc” basada en la agrupación temporal de personal (Clustering); finalmente, se presentan resultados de estudios preliminares en España.

En [Su and Lucko \(2016\)](#), los autores exploran un nuevo enfoque de la teoría del despliegue de múltiples equipos en proyectos de construcción; en primer lugar, se revisan las disimilitudes en la representación de la actividad, el punto de inicio y la medición de la velocidad y se explican sus causas gráfica y matemáticamente. En segundo lugar, las tripulaciones múltiples se expresan con funciones de singularidad en un modelo de tres partes: asignación de tripulación, utilización de la tripulación y horario lineal de la tripulación. El modelo se actualiza automáticamente cuando se modifican los parámetros para los análisis “qué pasa si”. En tercer lugar, los Métodos de Asignación Lineal (LSM) y Línea de Equilibrio (LOB) están unidos para que las tripulaciones

puedan ser optimizadas de manera conjunta.

En [Öztop et al. \(2017\)](#) se abordan un caso real del CSP que incluye diferentes tipos de tripulación en sistemas de transporte público de pasajeros, para lo cual proponen como método de solución un problema de programación táctica de tareas fijas basa en un modelo de programación binario que incluye un esquema de generación de desigualdades válidas que eliminan secuencias de tareas que exceden el tiempo máximo de trabajo diario de un miembro de la tripulación. El método es validado en casos reales donde se alcanzan soluciones efectivas en términos de tiempo computacional para casos que incluyen hasta 120 tareas.

Finalmente en [Ciancio et al. \(2018\)](#), presentan un enfoque integrado para resolver dos problemas de tareas de programación en compañías de transporte público. Los autores utilizan en el primer problema (Vehicle Scheduling Problem) un algoritmo goloso o Greedy para encontrar una solución inicial, la cual es mejorada mediante una estrategia de Recocido Simulado que explota un conjunto de técnicas de búsqueda local. Para el segundo problema (Crew Scheduling Problem), se usa una técnica clásica de asignación secuencial, la solución general es mejorada por medio de cambios de localización de los diferentes vehículos, de tal manera que e minimice la función objetivo propuesta. Los algoritmos y las técnicas de solución son diseñadas con base en requerimientos de la operación real de un sistema de Transporte Público.

[Liu et al. \(2019\)](#) Proponen un modelo extendido considerando los intereses de los pasajeros en tiempos de espera y el interés de los operadores del sistema de transporte público, debido a que el tiempo de espera genera un gran impacto en el pasajero, llevándolo a usar en menor o mayor frecuencia el transporte público. Los autores se basan en la sincronización de horarios integrados, programación de vehículos y el costo de espera de pasajeros, para lo cual, presentan un

modelo no cóncavo o convexo para lograr la reducción de tiempo. Debido a la complejidad de la función objetivo y su no convexidad, no es posible obtener un óptimo global para el problema, Por esta razón se utiliza un método de solución que combina el uso de las técnicas *Deficit method* (DF) con características gráficas para representar la cantidad de vehículos requeridos en cada estación, *Shifting Departure Time* (SDT) para modificar la ruta de inicio y por último el uso del *Deadheading* (DH) para reducir los tiempos de inactividad de los vehículos y reduce aún más el número de vehículos necesarios. El estudio es realizado con su sistema de prueba construido por los autores, basándose en tres importantes líneas de transporte público con alta demanda, recolectando información por un periodo de dos meses para determinar el flujo de pasajeros. Los resultados obtenidos muestra una mejora en la la reducción de buses necesarios para cumplir las rutas, y el tiempo de espera para los pasajeros logró mantenerse en el rango límite de 20 minutos.

Debido a lo anterior, es posible afirmar que el enfoque aquí propuesto brinda una alternativa de solución diferente a los trabajos anteriores. La metodología integrada es eficiente, dado que descompone el problema en subproblemas bastante conocidos y bien resueltos en la literatura.

Capítulo 3

Formulaciones matemáticas

Este trabajo busca resolver de manera articulada los problemas de programación de turnos de conductores, a partir de una adecuada asignación de servicios a los vehículos, particularmente, los esfuerzos de este estudio se enfocan en la programación de conductores. Este problema cuenta con una amplia literatura especializada, sin embargo, replicar las metodologías no genera resultados adecuados, a causa de las especificidades en los requerimientos de los operadores, diferencias en las regulaciones laborales del país donde se realiza cada estudio, y características puntuales del sistema de transporte donde se aplica.

Se requiere entonces un estudio detallado desde modelos generales de los problemas de asignación involucrados como punto de partida, para luego particularizar las formulaciones y metodologías de solución al caso de estudio. En las secciones [3.1](#) y [3.2](#) se presentan entonces de manera general, los problemas de programación de conductores y programación de vehículos respectivamente.

3.1. Formulación del problema de programación de conductores

De acuerdo a la revisión de la literatura del problema CSP existen diversas formulaciones matemáticas adaptadas a cada caso y a cada necesidad en particular, por lo tanto, no existe una formulación estándar capaz de resolver cualquier problema asociado a un CSP, sin embargo, con el fin de dar alcance a los objetivos planteados y por la similitud en restricciones con el problema planteado en este trabajo, la formulación matemática usada, corresponde a la realizada por [Chen and Niu \(2012\)](#), quienes a partir de un conjunto dado de viajes en un horizonte de planeación fijo, generan un conjunto factible de turnos de trabajo, los cuales incluyen tiempo de inicio, tiempo de finalización, tipo de servicio que pueden realizar los operadores, y una serie de viajes que cubren los horarios de cada una de las rutas planeadas con una frecuencia estipulada y satisface una serie de restricciones establecidas por las regulaciones laborales de cada país.

Los viajes que son asignados a un mismo operador constituyen la tarea que el operador debe desarrollar en una jornada laboral, y todo el conjunto de tareas constituyen la programación de la tripulación. Normalmente, el criterio usado para realizar la programación de una tripulación, consiste en el uso eficiente de los recursos (operadores) mientras se mantiene la integralidad de las reglas de trabajo. Los objetivos propuestos en las formulaciones matemáticas del CSP regularmente consisten en minimizar el costo de la tripulación, minimizar el número de operadores requeridos, maximizar la eficiencia de la tripulación, minimizar el tiempo de ocio de la tripulación entre otros. A continuación, se presenta la definición de la notación utilizada en el modelo matemático.

3.1.1. Notación matemática

- N Número de viajes dados en una tabla de trabajo para un día (*Timetable*).
- G Número de tipos de tareas de la tripulación.
- η Número de operadores disponibles en un día de trabajo.
- θ Número mínimo de operadores requeridos para realizar una tarea.
- i Índice de los viajes, $i = 1, 2, \dots, N$.
- k Índice de los operadores, $i = 1, 2, \dots, \theta$.
- g Índice del tipo de tarea.
- z_g Número de conductores asignados a la tarea g .
- d_i Tiempo de salida del viaje i en la estación de inicio.
- a_i Tiempo de llegada del i viaje a la estación final.
- T_0 Tiempo mínimo para intercambio en la estación de inicio por cada operador.
- T_g Máximo valor de Tiempos de trabajo por cada operador con la tarea g .
- B_g Tiempo de inicio más temprano de la tripulación con la tarea g .
- E_g Tiempo de inicio más tarde de la tripulación con la tarea g .
- x_{gk}^i Variable de decisión que, toma el valor de 1 si el viaje i es realizado por el operador en la tarea g , y toma el valor de cero en caso contrario.
- y_{gk}^{ij} Variable de decisión que, toma el valor de 1 si el viaje j se realiza inmediatamente después de realizar el viaje i por el operador k en la tarea g , y toma el valor de cero en caso contrario.

3.1.2. Modelo matemático

Matemáticamente, el problema de CSP puede ser representado el modelo binario de programación entera de cuatro índices que se representa por medio del conjunto de ecuaciones (1)-(8), el cual fue propuesto por [Chen and Niu \(2012\)](#).

$$\text{Min } Z = \sum_{g=1}^G \sum_{k=1}^{z_g} \sum_{i=1}^N \sum_{j=1}^N (d_j - a_i) y_{gk}^{ij} \quad (3.1)$$

s.a.

$$d_j x_{gk}^j - a_i x_{gk}^i \geq T_0 \quad \forall g, k \quad (3.2)$$

$$\sum_{i=1}^N x_{gk}^i (a_i - d_i) \leq T_g \quad \forall g, k \quad (3.3)$$

$$B_g \leq x_{gk}^i d_i \leq E_g \quad \forall g, k \quad (3.4)$$

$$z_g \geq \theta \quad (3.5)$$

$$\sum_{g=1}^G z_g \leq \eta \quad (3.6)$$

$$\sum_{g=1}^G \sum_{k=1}^{z_g} x_{gk}^i = 1 \quad \forall i \quad (3.7)$$

$$y_{gk}^{ij} = x_{gk}^i x_{gk}^j \quad \forall i, j, g, k \quad (3.8)$$

El objetivo del problema de programación de conductores consiste en maximizar el uso de todo el personal disponible y consecuentemente reducir el costo de operación; una de las formas de maximizar el uso del personal consiste en minimizar el tiempo de ocio, es decir, reducir en la medida de lo posible el lapso entre el fin de un viaje y el inicio del siguiente, tal como se representa en la función objetivo (3.1). El conjunto de restricciones (3.2) aseguran el cumplimiento

del tiempo de parada entre dos viajes adyacentes en la estación de inicio realizados por el mismo conductor. Las desigualdades (3.3) y (3.4) representan las restricciones de intensidad de trabajo y compatibilidad del tipo de tarea respectivamente. La restricción (3.5) asegura que el número de conductores con la tarea es mayor o igual al mínimo de conductores requeridos para dicha tarea. La desigualdad (3.6) corresponde a la restricción de máximo número de conductores disponibles. Las ecuaciones (3.7) asegura que cada viaje debe ser realizado exactamente por un único conductor. Finalmente, las ecuaciones (3.8) muestra la relación existente entre las variables de decisión.

3.2. Formulación del problema de asignación de vehículos con múltiples depósitos

El MDVSP puede ser representado a través de un conjunto T_1, T_2, \dots, T_n de tablas de programación (time tables) de viajes que deben ser realizados en un horizonte de tiempo τ . Cada viaje T_i está caracterizado tanto por el tiempo y lugar de inicio, así como el de finalización. Sea D_1, D_2, \dots, D_k el conjunto de depósitos y d_l el número de vehículos estacionados en el depósito D_l .

Una pareja de viajes (T_i, T_j) es compatible, si el viaje T_j puede ser realizado inmediatamente después de la finalización del viaje T_i por un mismo vehículo, esto es posible, si el tiempo de finalización del viaje T_i mas el tiempo de viaje en vacío (deadhead) desde T_i hasta T_j (sumados con un margen de seguridad suficiente), es menor que el tiempo de inicio del viaje T_j .

Así las cosas, el **objetivo** del MDVSP es conectar o agrupar parejas de viajes en cadenas, iniciando y terminando en el mismo depósito de vehículos, de esta manera, un vehículo es asignado

a cada uno de ellos de tal manera que la suma de los costos asociados a la operación sea mínimo.

La formulación propuesta en [Mesquita and Paixão \(1992\)](#), es una alternativa adecuada de implementación, principalmente porque no presenta conjuntos de restricciones con explosión combinatorial.

En dicha formulación, se establece un modelo matemático basado en teoría de grafos para el MDVSP. Considerando un grafo $G = (V, A)$ definido de manera tal que, un conjunto de vértices $V = \{1, 2, \dots, n+k\}$ es particionado en $I = \{1, 2, \dots, m\}$, correspondiente a los viajes T_1, T_2, \dots, T_n , y $D = \{n+1, n+2, \dots, n+k\}$, correspondiendo a los depósitos D_1, D_2, \dots, D_k . El conjunto A contiene los arcos (i, j) correspondiente a las parejas de viajes factibles (T_i, T_j) y todos los arcos de salidas y entradas a los depósitos $(i, n+l), (n+l, i)$ para $l = 1, 2, \dots, k$ e $i = 1, 2, \dots, n$. Un costo c_{ij} está asociado a cada arco de manera que, para cada $(i, j) \in I \times I$, dicho costo estima el consumo de combustible necesario para el viaje tipo deadhead desde i hasta j , y otras penalidades que quiera aplicar la compañía de transporte. El costo relativo a los enlaces desde o hacia un depósito se denotan como: $c_{n+l, n+l} = 0, l = 1, 2, \dots, k$ y $c_{n+l, n+l'} = \infty, l, l' = 1, 2, \dots, k, l \neq l'$.

En el caso particular en que las compañías de transporte busquen minimizar de manera simultánea el tamaño de la flota empleada y los costos de viajes en vacío (deadheads), entonces $c_{n+l, i}(c_{i, n+l})$ representa el costo del viaje en vacío desde el depósito D_l al lugar de inicio del viaje T_i (desde el lugar de finalización del viaje T_i al depósito D_l), mas la mitad del costo incurrido por la utilización del vehículo perteneciente al depósito D_l .

Ahora, el MDVSP consiste en encontrar el conjunto de circuitos de mínimo costo que cumplan las siguientes condiciones: (i) Cada vértice $i \in I$ es cubierto exactamente una vez por un circuito, (ii) Cada circuito contiene exactamente un vértice del conjunto D y (iii) El número de circuitos que

cubren el vértice $n + l$ nunca excede $d_l (l = 1, 2, \dots, k)$.

3.2.1. Definición de variables de decisión

Las variables de decisión del modelo matemático se especifican a continuación:

$$x_{ij} = \begin{cases} 1, & \text{si el viaje } i \text{ está directamente conectado con el viaje } j \\ 0, & \text{de lo contrario} \end{cases} \quad (3.9)$$

$$x_{i,n+l} = \begin{cases} 1, & \text{si después del viaje } i \text{ el bus retorna al depósito } D_l \\ 0, & \text{de lo contrario} \end{cases} \quad (3.10)$$

$$x_{n+l,j} = \begin{cases} 1, & \text{si el depósito } D_l \text{ provee un bus para atender el viaje } j \\ 0, & \text{de lo contrario} \end{cases} \quad (3.11)$$

$$y_{n+l,j} = \begin{cases} 1, & \text{si el viaje } i \text{ es realizado por un vehículo del depósito } D_l \\ 0, & \text{de lo contrario} \end{cases} \quad (3.12)$$

3.2.2. Formulación matemática

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} + \sum_{l=1}^k \left(\sum_{i=1}^n c_{i,n+l} x_{i,n+l} + \sum_{j=1}^n c_{n+l,j} x_{n+l,j} \right) \quad (3.13)$$

s.a

$$\sum_{j=1}^{n+k} x_{ij} = 1 \quad i = 1, 2, \dots, n \quad (3.14)$$

$$\sum_{j=1}^{n+k} x_{ij} = 1 \quad i = 1, 2, \dots, n \quad (3.15)$$

$$\sum_{j=1}^n x_{n+l,j} \leq d_l \quad l = 1, 2, \dots, k \quad (3.16)$$

$$x_{n+l,j} - y_{jl} \leq 0 \quad j = 1, 2, \dots, n; l = 1, 2, \dots, k \quad (3.17)$$

$$x_{i,n+l} - y_{il} \leq 0 \quad i = 1, 2, \dots, n; l = 1, 2, \dots, k \quad (3.18)$$

$$y_{il} + x_{ij} - y_{jl} \leq 1 \quad i, j = 1, 2, \dots, n; l = 1, 2, \dots, k \quad (3.19)$$

$$\sum_{l=1}^k y_{il} = 1 \quad i = 1, 2, \dots, n \quad (3.20)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, n+k \quad (i, j) \neq (n+l, n+l') \quad l, l' = 1, 2, \dots, k \quad (3.21)$$

$$y_{il} \in \{0, 1\}, \quad i = 1, 2, \dots, n; l = 1, 2, \dots, k \quad (3.22)$$

$$\sum_{l=1}^k \sum_{j=1}^n x_{n+l,j} \leq UB_{Vehículos} \quad (3.23)$$

Los conjuntos de restricciones controlan los siguientes aspectos:

- El conjunto (3.14) asegura que para cada $i = 1, 2, \dots, n$; al completar el viaje i , el bus retorna

a un depósito o inicia otro viaje $j \in I$.

- Las restricciones en (3.15) imponen, para cada $j = 1, 2, \dots, n$, la asignación del bus a un viaje j , bien sea desde el depósito o desde otro viaje.
- En (3.16) se garantiza, para $l = 1, 2, \dots, k$ que la capacidad d_l de un depósito no sea violada.
- (3.17) implica que para cada par j y l , la asignación entre el viaje j y el depósito D_l , siempre que el viaje i es el primer viaje de una cadena realizada por un vehículo que inicia su servicio desde el depósito D_l .
- (3.18) asegura que para cada par i y l , la asignación entre el viaje i y el depósito D_l , siempre que el viaje i es el último viaje de una cadena realizada por un vehículo que retorna al depósito D_l .
- (3.19) establece que para cada par $(i, j) \in I$, si un viaje i está conectado directamente con el viaje j y el viaje i es asignado al depósito D_l , entonces, el viaje j es asignado al depósito D_l .
- (3.20) garantiza que cada viaje $i \in I$ es asignado a exactamente un depósito.

Capítulo 4

Metodología

Este proyecto presenta un alto componente de investigación aplicada, y por esta razón, el problema ha sido abordado en primera instancia, tan cercano como ha sido posible al caso real. Las metodologías han sido elegidas para responder progresivamente al total de particularidades de la operación del sistema de transporte masivo objetivo.

En la primera etapa se propone un enfoque heurístico, como se presenta en la subsección [4.1](#). Esta aproximación fue elegida en la etapa temprana de la investigación, dada la flexibilidad que presenta para incluir detalles técnicos de la asignación de conductores, además de los tiempos de respuesta, adecuados para el centro de control del operador. La heurística, al incluir recomendaciones de funcionarios del sistema expertos, consigue la disminución de algunos turnos de trabajo, en tiempos de cómputo muy bajos.

Así las cosas, el desarrollo de la heurística permitió un levantamiento detallado de los requerimientos del operador alrededor de los turnos, parte fundamental en los estudios de asignación de conductores como se ha mencionado. Naturalmente, el enfoque heurístico presenta

limitaciones, y se concentra únicamente en la disminución del personal requerido, sin buscar necesariamente una carga laboral equilibrada entre los turnos resultantes. La inclusión de este balance, incrementa sustancialmente la complejidad del problema, por lo tanto, se requiere de una metodología capaz de integrar este criterio con la disminución de costos. En la subsección 4.2, se presenta el modelo matemático propuesto a la medida de la operación del sistema, teniendo en cuenta la minimización del personal y la carga laboral de los turnos. El nuevo modelo, es resuelto mediante generación de columnas, aproximación especializada en problemas de asignación, donde el alto grado de infactibilidad en sus espacios de solución es natural.

Las metodologías 4.1 y 4.2 fueron validadas por el centro de control del operador del sistema, presentando una disminución satisfactoria en los costos asociados. Ahora bien, para consolidar dichas metodologías, es necesario probarlas en instancias de la literatura especializada, permitiendo tener un punto de comparación para cuantificar su rendimiento y depurar cualquier inconsistencia. Las librerías publicadas, presentan aleatoriedad y variedad dimensional necesarias para poner a prueba las propuestas de solución.

Dado que los estudios de programación de conductores suelen ser trabajos de investigación aplicada, los casos de prueba normalmente son información reservada. Después de una exploración importante de la literatura para obtener casos de referencia, se identificó el repositorio presentado en [Beasley et al. \(1996\)](#), el cual ofrece una librería de instancias generadas aleatoriamente, siguiendo un procedimiento similar al propuesto en [Fischetti et al. \(1987\)](#), referente de construcción de instancias en varios trabajos de asignación de conductores. Este repositorio presenta además un nivel de generalidad suficiente para tener un punto de encuentro entre las metodologías de este trabajo, e indicadores de rendimiento. En la subsección 4.2.2 se desarrolla el

estudio de las instancias de prueba, empleando el modelo propuesto por [Beasley et al. \(1996\)](#). Este último, deja abiertos algunos aspectos técnicos, requiriendo la construcción de un nuevo modelo que permita encontrar las soluciones reportadas, y de esta manera, llevar a cabo la validación de las metodologías.

En cuanto al problema de asignación de vehículos, se integró la metodología desarrollada en [Moreno et al. \(2019\)](#), investigación aplicada al mismo sistema de transporte masivo de este trabajo, la cual hace parte de la literatura especializada. En la Figura 4.1 se muestran los componentes de dicha metodología, una propuesta híbrida que explota los beneficios de un algoritmo poblacional como es el algoritmo genético [Chu and Beasley \(1998\)](#), generando soluciones diversas y de buena calidad para alimentar un modelo tipo Set Partitioning [Subramanian et al. \(2013\)](#), generando una solución inicial al modelo presentado en 3.2.

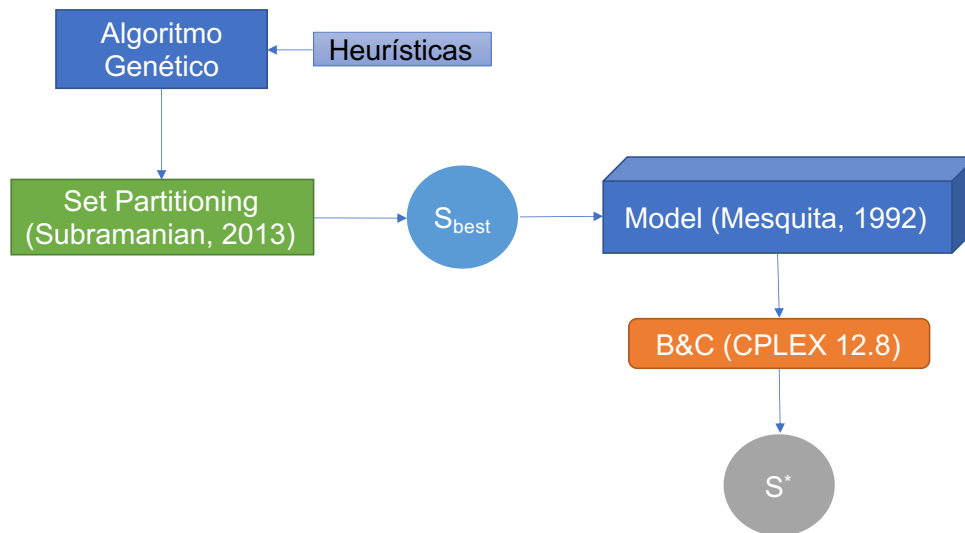


Figura 4.1: Metodología híbrida de solución problema de programación de vehículos MDVSP. Tomado de [Marín-Moreno \(2019\)](#).

Como se presenta en el marco referencial, en la Figura 2.1, el enfoque de etapas es el recomendado por la literatura especializada, por tal razón, las etapas de la metodología suceden

como se presenta en la Figura 4.2, donde se resaltan las soluciones en cascada del MDVSP y el CSP, enfatizando en la programación de conductores, que asocia temas de gestión de personal muy particulares de la cultura organizacional, regulación laboral y competitividad del sector.

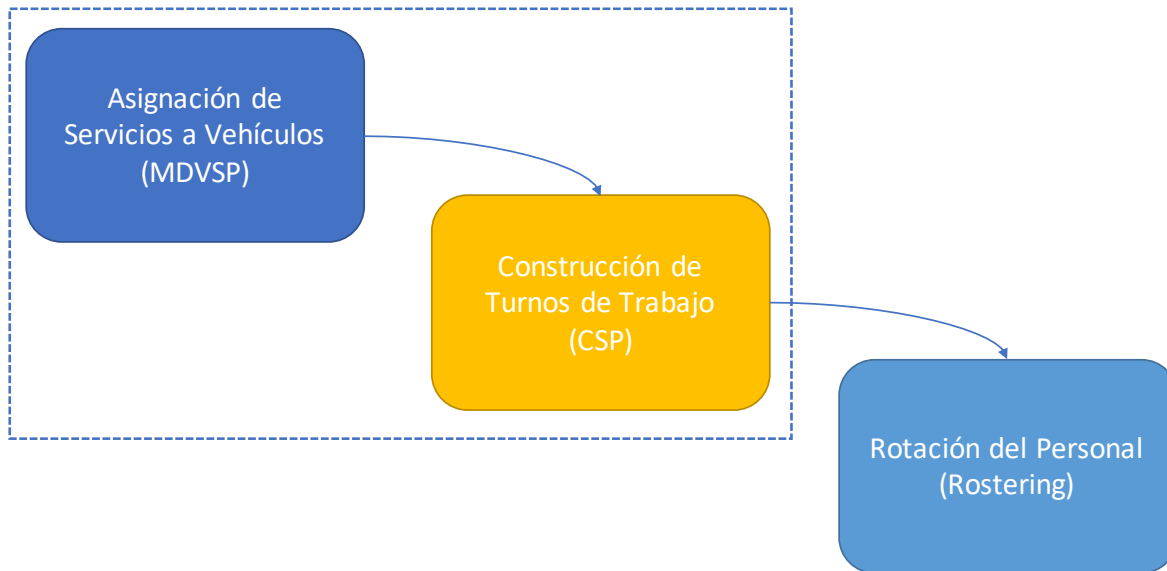


Figura 4.2: Metodología híbrida de solución problema de programación de vehículos MDVSP. Fuente: Elaboración propia.

4.1. Enfoque heurístico

El problema de CSP en la operación de Sistemas de Transporte Masivo consiste en generar turnos de trabajos para sus operadores de tal manera que se realicen todos los viajes programados de acuerdo a frecuencias establecidas por el ente gestor de dichos sistemas. El problema inicia a partir de diferentes tablas de servicios (*Timetables*) otorgadas por el ente gestor, que para efectos de este trabajo es [Megabús](#), quien se encarga de entregar la programación de viajes que se deben realizarse diariamente en el Área Metropolitana Centro Occidente; las tablas asignadas por parte de Megabús deben ser cubiertas en su totalidad por las empresa Operadora de Transporte Masivo

Integra S.A.; el problema para la empresa operadora, radica en que cada una de las tablas son de tamaños diversos (en tiempo y cantidad de viajes) dependiendo de la Tiempo la ruta y la frecuencia, esto implica que, normalmente la cantidad de viajes que posee cada tabla no pueden ser realizados por un único conductor, ya que existen regulaciones laborales tanto internas como externas que estipulan un tiempo máximo de trabajo continuo, tiempo máximo de trabajo por día, tiempo mínimo de descanso después de realizar un turno de trabajo entre muchas otras restricciones, es por ello que, surge la necesidad de realizar turnos de trabajo adecuados, para lo cual es necesario realizar la partición de cada tabla de servicios, de tal forma que cada una de las particiones coincidan con las regulaciones de trabajo expuestas.

En la Figura 4.3 se describe de forma general la metodología aquí propuesta, la Figura 4.3-(a) muestra que el proceso inicia con un conjunto de tablas de programación de servicios que se deben realizar (*Timetables*) diariamente. La Figura 4.3-(b) ilustra el proceso de división de las tablas en sub-tablas, las cuales cumplen con un conjunto de restricciones técnicas de trabajo. Finalmente, la Figura 4.3-(c) muestra un de emparejamiento donde 2 o más sub-tablas pueden ser unidos para formar un turno de trabajo de un operador durante un día y que cumpla con todas las restricciones técnicas de operación y regulación laboral.

Para solucionar el problema de CSP de Integra S.A. se propone un método híbrido de dos fases; en la primera fase se realiza la división de cada una de las tablas de servicios acorde a el tiempo máximo de trabajo continuo permitido, para lo cual se realiza una adaptación del método de división de secuencias propuesto en Prins (2004) y extendido en Liu et al. (2009), mientras que en la segunda fase se realiza una adaptación del método de presentado por Kruskal para árboles de expansión mínima Jungnickel (2008), por el medio del cual, se realiza un emparejamiento entre

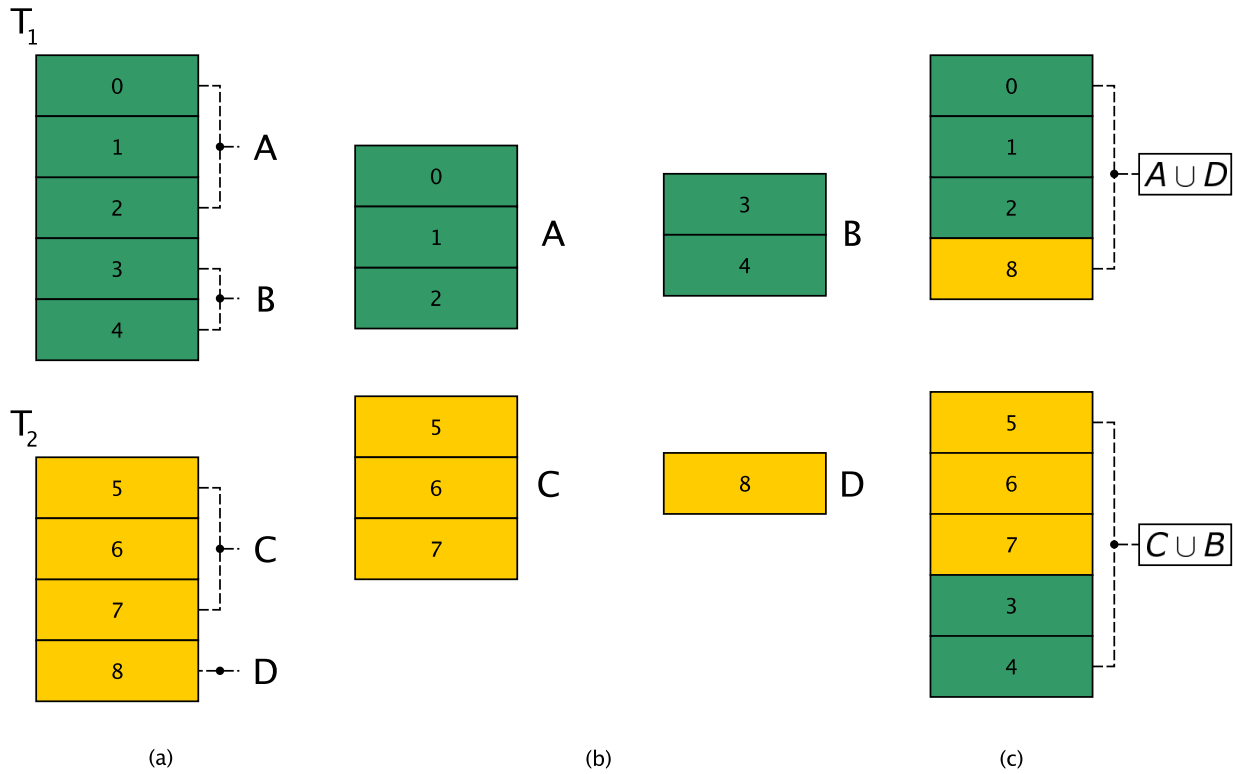


Figura 4.3: Metodología general de CSP para Integra S.A. Fuente: Elaboración propia.

sub-tablas que cumplan con las restricciones técnicas y laborales.

El Algoritmo 1 muestra de manera general, la metodología propuesta para solucionar el problema de CSP aplicado al caso real de la empresa Integra S.A. operadora del Sistema de Transporte Masivo del Área Metropolitana Centro Occidente.

4.1.1. División de tablas

De acuerdo al orden de atención brindado por una secuencia de viajes, que en este caso se establece con el orden ascendente de los tiempos de inicio de cada uno de los viajes, se obtiene el subgrafo auxiliar. En la Figura 4.4 se muestra el mencionado subgrafo.

En la Figura 4.4, se muestran conexiones entre todos los viajes de la secuencia en el subgrafo, además no es posible por ejemplo atender el viaje T_2 después del viaje T_1 , o el viaje T_6 después

Algorithm 1 Constructivo SecuenciaPrins+KruskalModificado

Data: N_h - Número de horas de un bloque de trabajo, T_b - Tiempo de descanso, $T_{d_{ij}}$ Tiempo de desplazamiento entre los puntos de finalización del bloque de trabajo i y el punto de inicio del bloque de trabajo j , C_r - Conjunto de restricciones para formar turnos de trabajo, N_t - Número máximo de bloques de trabajo para constituir un turno, N_{C3} - Número de conductores con licencia C3, N_{C2} - Número de conductores con licencia C2

Result: Turnos de trabajo Caso Integra S.A.

$Viajes = ProcesarRegistros(BD_Eventos);$

for $Viajes_{Ruta_i} \in Viajes$ **do**

$BloquesTrabajo_{Ruta_i} = SecuenciaPrins(Viajes_{Ruta_i}, N_b);$

$BloquesTrabajo \leftarrow BloquesTrabajo \cup BloquesTrabajo_{Ruta_i};$

end

$GrafoBloquesTrabajo \leftarrow \emptyset;$

for $BloqueTrabajo_i \in BloquesTrabajo$ **do**

$nodo_i = construirNodo(BloqueTrabajo_i);$

$GrafoBloquesTrabajo.nodos \leftarrow GrafoBloquesTrabajo.nodos \cup nodo_i;$

end

for $nodo_i \in GrafoBloquesTrabajo.nodos$ **do**

for $nodo_j \in GrafoBloquesTrabajo.nodos$ **do**

if $nodo_i \neq nodo_j \wedge C_r$ **then**

$GrafoBloquesTrabajo.aristas \leftarrow GrafoBloquesTrabajo.aristas \cup (nodo_i, nodo_j)$

end

end

end

$N_t \leftarrow 3;$

$TurnosTrabajo \leftarrow KruskalModificado(N_t, N_{C3}, N_{C2}, GrafoBloquesTrabajo);$

return $TurnosTrabajo;$

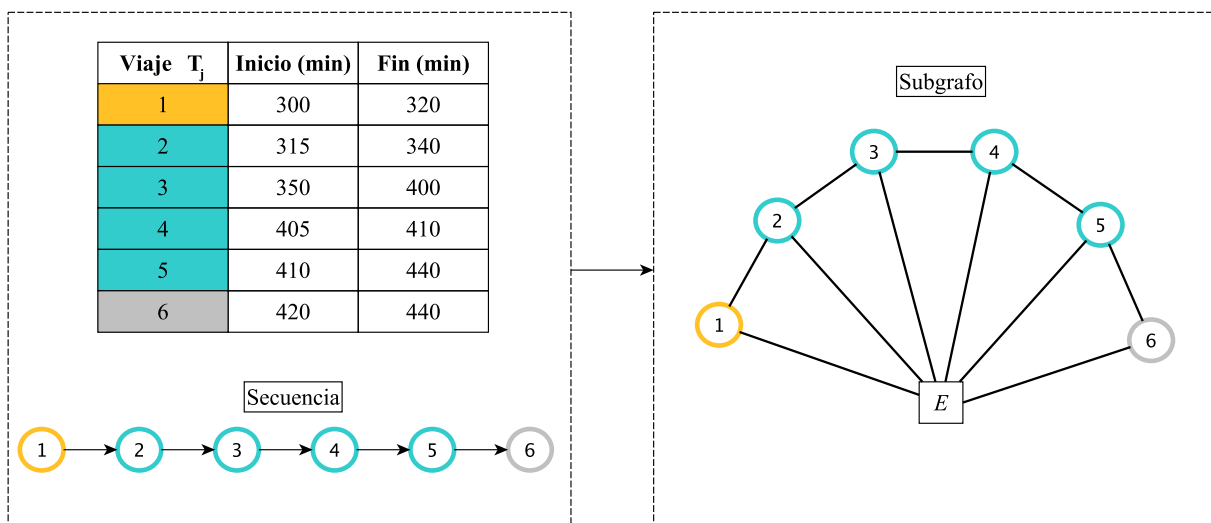


Figura 4.4: Subgrafo según la secuencia. Fuente: Elaboración propia.

del viaje T_5 porque hay traslape en los tiempos de inicio y finalización de los mismos. Estas conexiones se deben descartar para que no sean incluidas en las combinaciones que se explorarán en la siguiente fase. El subgrafo resultante se observa en la Figura 4.5.

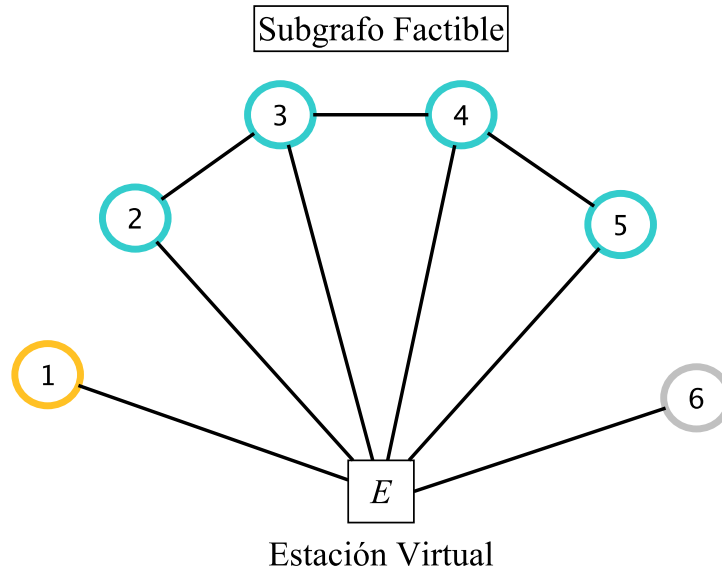


Figura 4.5: Subgrafo sólo con conexiones factibles entre viajes. Fuente: Elaboración propia.

Posteriormente se construye un grafo acíclico $H = (X, A)$ con $n + 1$ nodos, de esta manera X contiene un nodo estático o ficticio E más h vértices que representa el conjunto de viajes de una tabla en el orden establecido por el tiempo de inicio de cada uno de ellos $TT_j = T_1, T_2, \dots, T_h$. Seguidamente se crea un conjunto de aristas A , donde cada componente de dicho conjunto hace referencia a un arco (i, j) , es decir, que un número determinado de viajes contiguos pueden ser atendidos por un mismo turno, lo cual se cumple si, la suma de todos los tiempos de duración de los viajes incluidos en el arco no supera el tiempo máximo de trabajo continuo, para lo cual se debe cumplir la restricción (3.3) del modelo matemático.

El valor t_{ij} de cada arista, está dado por la sumatoria de la diferencia de tiempos que se obtiene al restar el tiempo máximo de trabajo para un turno o tarea denominado τ y la sumatoria del tiempo

de cada uno de los viajes incluidos en un determinado arco, tal como se muestra en la ecuación (4.1).

$$t_{ij} = T_g - \sum \text{tiempo de cada viaje} \quad (4.1)$$

La Figura 4.6 muestra el digrafo o grafo acíclico resultante. En este se muestra cómo se obtienen los valores t_{ij} y las aristas que surgen según las conexiones del subgrafo de la Figura 4.5.

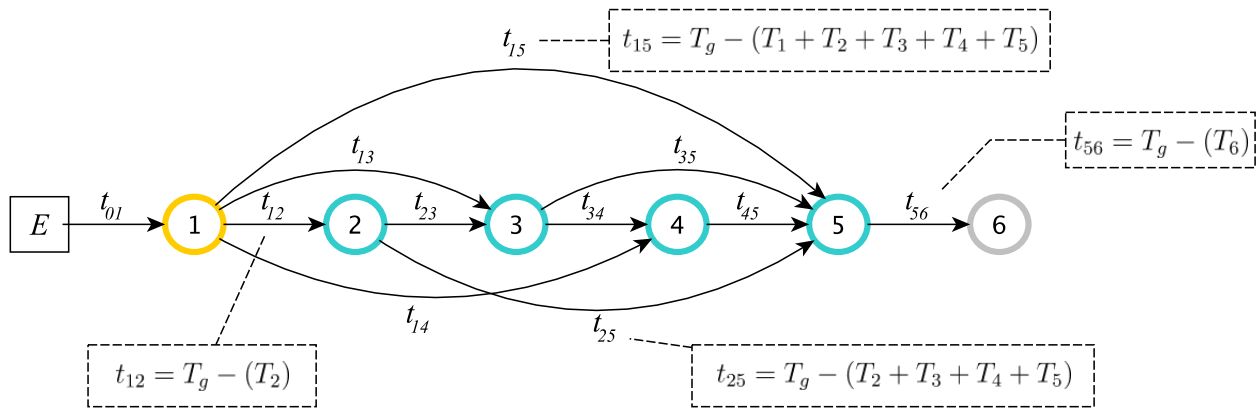


Figura 4.6: Digrafo para encontrar una partición balanceada de una ruta. Fuente: Elaboración propia.

Una vez se tienen todos los arcos factibles con respecto al tiempo máximo de trabajo continuo, se procede a determinar la partición de cada tabla de programación de viajes que superan el tiempo máximo de trabajo continuo, para lo cual, se resuelve de manera exacta un problema de programación lineal para hallar el camino más corto entre dos puntos de una red.

Modelo matemático para el problema de la ruta más corta

Este modelo permite determinar el camino más corto entre dos vértices cualquiera de una red. Sea n el conjunto de nodos de una red, en la cual ingresa una unidad externa de flujo en el nodo s y sale en el nodo t , siendo s y t los nodos entre los cuales se desea encontrar la ruta más corta, también se define a x_{ij} como la cantidad de flujo en el arco (i, j) y c_{ij} como el costo del arco (i, j) . El modelo matemático está descrito por las ecuaciones (4.2) a (4.4).

$$\text{Min } Z_2 = \sum_{i=1}^n \sum_{j=1}^n c_{ij}x_{ij} \quad (4.2)$$

s.a.

$$\sum_{j=1}^n x_{ij} - \sum_{k=1}^n x_{ki} = \begin{cases} 1; & \text{si } i = s \\ -1; & \text{si } i = t \\ 0; & \text{en otro caso} \end{cases} \quad \forall i = 1, 2, \dots, n \quad (4.3)$$

$$x_{ij} \in \{0, 1\} \forall i, j = 1, 2, \dots, n \quad (4.4)$$

La ecuación (4.2) representa la función objetivo, dado por el costo de las aristas utilizadas en la ruta que va desde el vértice s al vértice t . El conjunto de ecuaciones (4.3) representa la conservación del flujo en cada una de las aristas y finalmente la ecuación (4.4) representa el conjunto de variables binarias que representan el flujo por cada arista, las cuales toman el valor de uno si el flujo atraviesa la arista (i, j) y cero en caso contrario.

Los arcos más atractivos para establecer la ruta más corta entre el primero y el último nodo, son aquellos que incluyan más viajes. Para el ejemplo presentado, la ruta más corta se resalta en la

Figura 4.7.

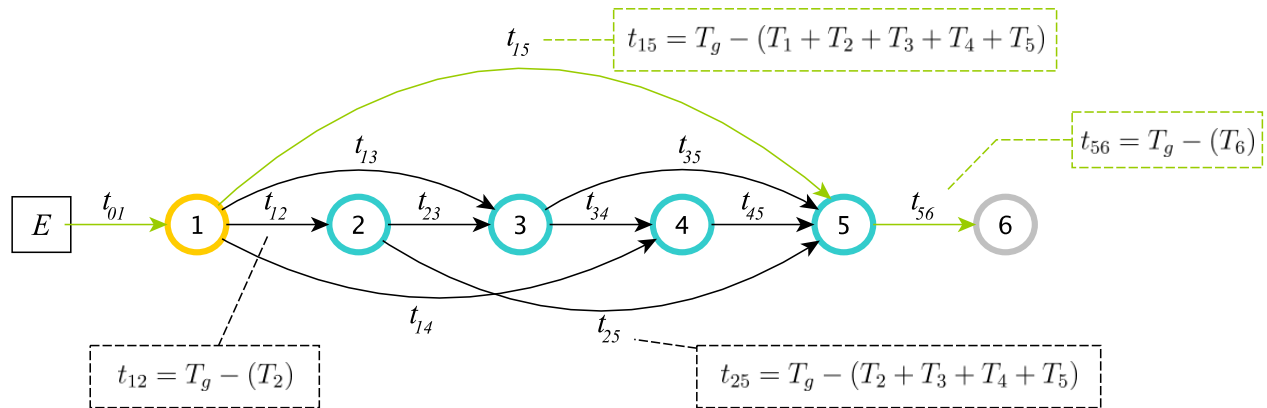


Figura 4.7: Ruta más corta que corresponde a una partición balanceada de la tabla de servicios. Fuente: Elaboración propia.

Los arcos solución, resaltados en la Figura 4.7, representan bloques de trabajo o itinerarios de atención de viajes. En el ejemplo ilustrado correspondería a la forma de atención de la Figura 4.8.

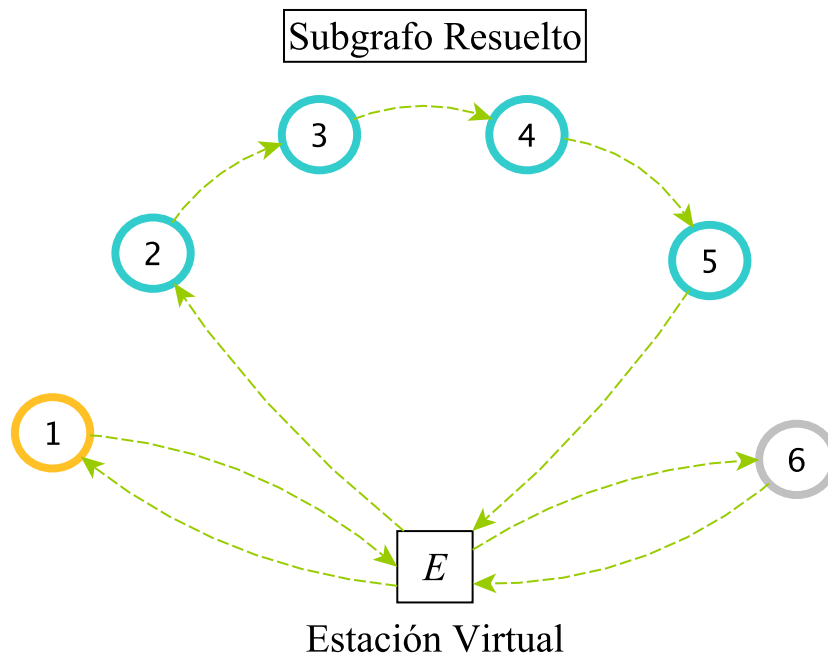


Figura 4.8: Agrupación de viajes en bloques de trabajo. Fuente: Elaboración propia.

4.1.2. Construcción de turnos de trabajo

Una vez se tienen los bloques de trabajo (grupos de viajes que se atienden de manera consecutiva), se procede a la fase de construcción de turnos. Los bloques de trabajo en esta fase estarán representados como nodos en el grafo que se presenta en la Figura 4.9. Esta última, busca ilustrar el Algoritmo de Kruskal y la modificación que se realiza en este trabajo para la construcción de jornadas de los conductores.

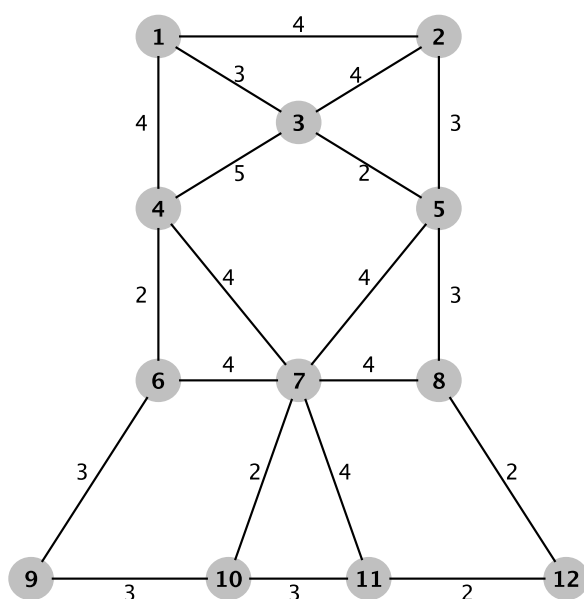


Figura 4.9: Grafo ilustrativo Algoritmo Kruskal. Fuente: Elaboración propia.

El Algoritmo de Kruskal, descrito en Jungnickel (2008), tiene como objetivo encontrar el árbol de expansión mínima en un grafo (MST, del inglés *Minimum Spanning Tree*). Esta estrategia para hallar el MST se ajusta adecuadamente al problema que se trabaja en esta investigación, dado que los grafos de programación y asignación de tareas son en mayor o menor medida incompletos. En la Figura 4.10 se muestra el MST obtenido con el Algoritmo de Kruskal del grafo ejemplo de la Figura 4.9.

Dado que el Algoritmo de Kruskal toma todos los nodos del grafo como islas que se van

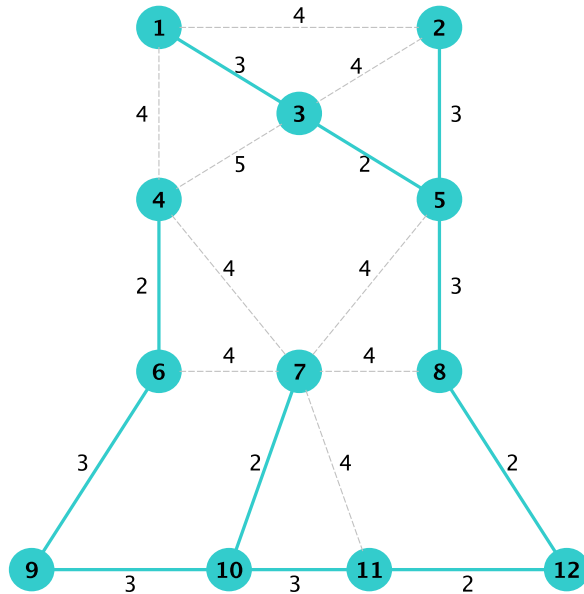


Figura 4.10: Árbol de expansión mínima. Fuente: Elaboración propia.

interconectando a través de las aristas de menor costo, con el fin de construir caminos de mínimo costo, la modificación que se realiza en este trabajo al Algoritmo de Kruskal, consiste en restringir la interconexión de caminos que están en crecimiento. En la Figura 4.11 se presenta una fase intermedia del Algoritmo de Kruskal aplicado al grafo de la Figura 4.9 y es en este punto donde el Algoritmo debe detenerse para el caso aquí presentado, porque se busca obtener el número mínimo de caminos de mínimo costo.

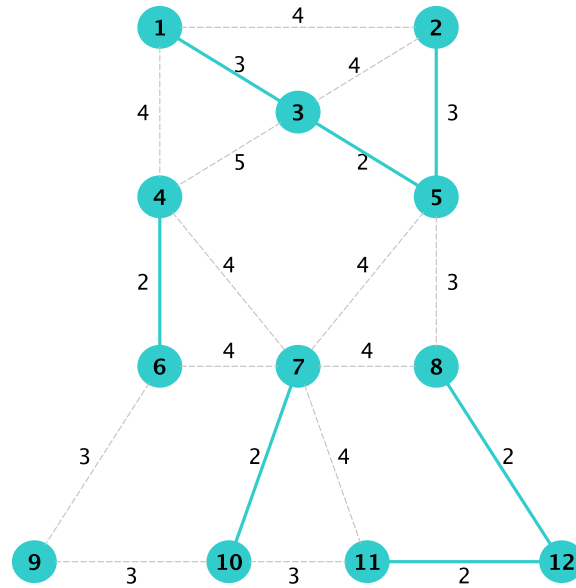


Figura 4.11: Kruskal modificado (caminos mínimos). Fuente: Elaboración propia.

Adicionalmente, para el caso real, los nodos pueden consumir dos recursos diferentes, que en este caso son conductores con licencia C3 (que son más costosos) y conductores con licencia C2. Si un camino o jornada de trabajo presenta un nodo que requiera licencia C3, entonces es necesario emplear un conductor de este tipo para que atienda la jornada. Por lo tanto, el Algoritmo de Kruskal Modificado, busca interconectar primero nodos que requieran licencia C2 para utilizar el mínimo número de conductores de licencia C3. En la Figura 4.12 se presenta una posible construcción de turnos de trabajo por el Algoritmo de Kruskal Modificado.

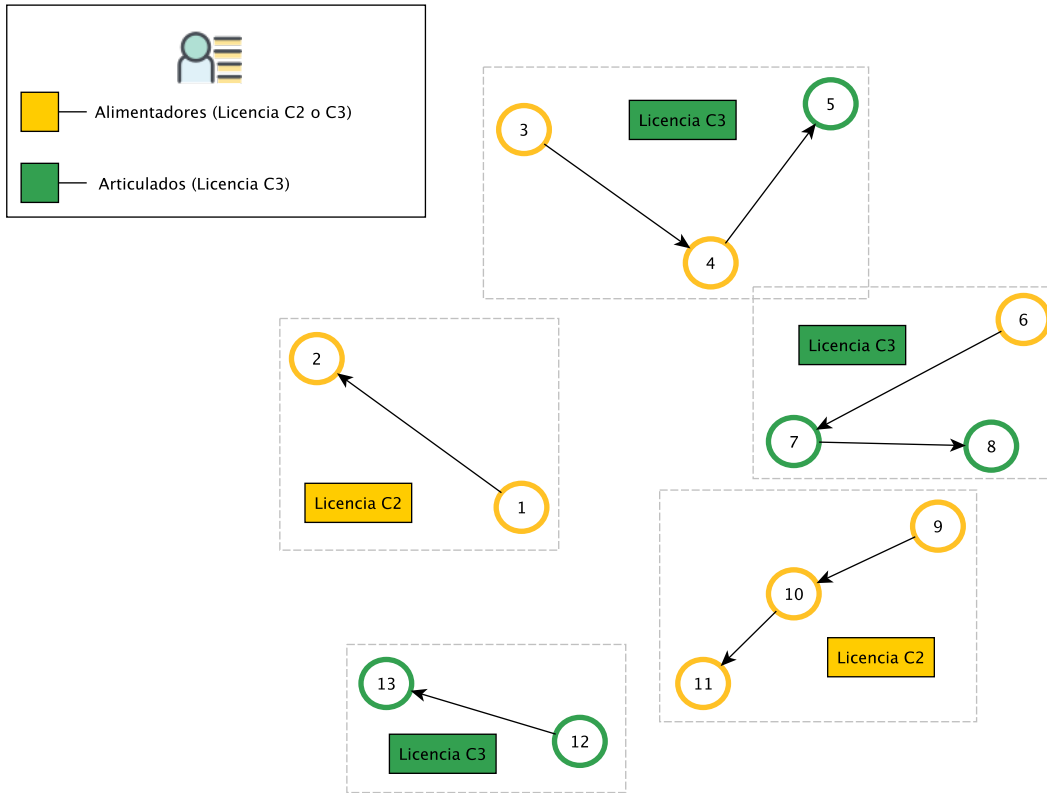


Figura 4.12: Kruskal modificado construcción turnos. Fuente: Elaboración propia.

4.1.3. Resultados enfoque heurístico

La metodología aquí propuesta, se implementó con el fin de encontrar soluciones de buena calidad en tiempos de computo mínimos, pensando en la programación real de los turnos de trabajo de los operadores de la empresa Integra S.A., operadora del Sistema de Transporte Masivo del AMCO (Área Metropolitana de Centro Occidente), la cual, en la actualidad debe atender un promedio de 4600 viajes diarios. La metodología fue implementada en el ambiente de programación C++®, bajo sistema operativo Linux distribución Ubuntu 14.04 LTS de 64 bits junto con el solver CPLEX® Versión 12.6 de 64 bits, un procesador Intel® Core™ i7-3630QM CPU @ 2.40GHz × 8 y memoria RAM de 8 GB.

Para validar entonces el enfoque presentado en esta sección, la empresa Integra S.A. seleccionó 3 casos de prueba de especial interés: (i) por su alto grado de dificultad, y (ii) porque ofrecen la variación de servicios que ha tenido que cubrir el operador, antes y después de encargarse del 100 % del sistema de transporte del AMCO. Se debe realizar entonces la programación diaria para 2338 servicios, correspondientes al 50 % del sistema en los días hábiles del año 2014; 4578 y 4608 viajes, correspondientes al 100 % del sistema en días hábiles durante los años 2019 y 2020 respectivamente.

En la Tabla 4.1 se muestran los resultados obtenidos de la programación diaria de turnos de trabajo en cada uno de los casos de la empresa Integra S.A.. Dicha tabla, muestra el nombre o código del caso de prueba, el tamaño en números de viaje de cada caso, la solución obtenida en turnos, tanto por tipo de conductor como el total de turnos generados y el tiempo computacional requerido para obtener cada solución. Puntualmente, la primera fila, denotada como *Operación Actual*, corresponde a la operación previa a la pandemia del virus SARS-COV 2, escenario más complicado para la metodología, al tratarse de un funcionamiento sin ninguna restricción, y por tanto, con un elevado número de servicios. La segunda fila, denotada como *Caso Prueba 1 (28122014_122430)*, corresponde a la operación del año 2014, donde la empresa era responsable únicamente del 50 % del sistema BRT. Finalmente, *Caso Prueba 2 (31082016_090004)*, presenta el mayor número de servicios durante una contingencia presentada en el Área Metropolitana, sumando temporalmente servicios de transporte urbanos accesorios que no son responsabilidad permanente del operador.

Para la toma de decisiones sobre la asignación de los servicios, la empresa dispone de expertos en el área de operaciones, que con sus 20 años de experiencia en los sistemas de transporte del

Nombre del Caso	del No. de Servicios	Programación Manual			Algoritmo			Tiempo Ejecución (segundos)
		Turnos C2	Turnos C3	Total Turnos	Turnos C2	Turnos C3	Total Turnos	
Operación Actual	4578	62	69	131	63	66	129	18,50
Caso Prueba 1 (28122014_122430)	2338	33	29	62	36	25	61	1,47
Caso Prueba 2 (31082016_090004)	4608	63	70	133	63	69	132	27,82

Tabla 4.1: Comportamiento de la metodología en los casos de operación de Integra S.A.

AMCO, establecen manualmente la programación de los turnos de trabajo para cada conductor.

En general, este proceso tarda alrededor de una semana en realizarse, específicamente, una jornada laboral completa, y una serie de ajustes en los días sucesivos.

Nótese en la Tabla 4.1, los buenos resultados obtenidos, tratándose incluso de un enfoque heurístico. Se resaltan los valores de las columnas **Turnos C3** y **Total Turnos**, correspondientes al **Algoritmo** en cada uno de los casos, porque son los indicadores que más interesan en este problema. Obsérvese que:

1. Los turnos que requieren licencia C3 por parte de los operadores, involucran servicios para vehículos articulados, es decir, conductores más calificados, y por tanto, un pago más elevado por cada hora trabajada. Disminuir el número de conductores necesarios con esta licencia para la realización de los servicios, impacta positivamente los costos de operación. Este valor es más elevado en la programación generada manualmente por los expertos, dado que no controlan tanto como puede hacerlo la metodología, la construcción de turnos mixtos, es decir, turnos que incluyen conducción de vehículos alimentadores y vehículos articulados. Se invita al lector a comparar en cada caso la columna **Turnos C3** del **Algoritmo** con la columna correspondiente en **Programación Manual**. En todos los casos, el valor es superado por la

metodología, es decir, es menor.

2. El total de turnos es quizás el indicador que más interesa al operador, porque impacta en mayor medida los costos de operación del sistema. Nuevamente, en todos los casos, la columna **Total Turnos** para el **Algoritmo**, es menor que su homóloga para la **Programación Manual**.
3. Los tiempos de cómputo presentados en la Tabla 4.1, en todos los casos es notablemente menor, supera la semana de trabajo que requiere el centro de operaciones para ajustar los turnos de sus tripulantes .

Resultados satisfactorios para un enfoque que puede mejorarse con aleatorización controlada (extender a metaheurística) o hibridación matemática, muestran la relevancia del aporte, permitiendo mejorar de manera inmediata la operación de la empresa. Finalmente, los tiempos de cómputo requeridos para cada caso, muestran que a pesar de que el número de servicios genera un crecimiento exponencial en las posibilidades de solución, son estables en comparación con el tamaño del problema: *Operación Actual* y *Caso Prueba 2 (31082016_090004)* requieren entre 20 y 25 segundos de cómputo con una media de 4600 servicios, mientras que *Caso Prueba 1 (28122014_122430)* con 2338 servicios, genera una solución adecuada en menos de 2 segundos de tiempo de cómputo.

4.2. Enfoque matemático

Se propone una metodología de solución, en donde se generan turnos a partir de la selección deseada de las tablas, es decir, el problema de Programación de Conductores CSP (Del inglés:

Crew Scheduling Problem), sobre los bloques de trabajo resultantes de la partición adecuada de las tablas de servicios (En inglés: *Timetables*), establecidas por el ente gestor al operador. El modelo propuesto posteriormente buscará generar turnos con el conjunto de porciones de tabla de entrada que se le ingresen como parámetros, combinando sin restricción las rutas que se encuentren. Es decir, que si se pretende no mezclar las rutas de Troncal con las de Alimentador, el procedimiento será usar el modelo ingresando únicamente las tablas de Troncal, y posteriormente ingresar el resto de tablas de trabajo de Alimentador, como se señala en la Figura 4.13.

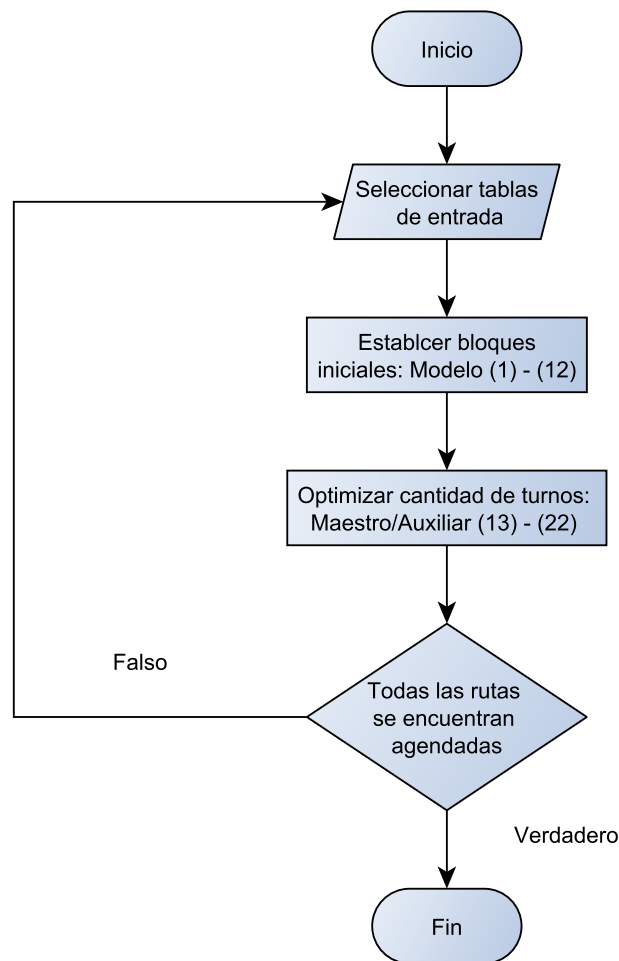


Figura 4.13: Diagrama de flujo algoritmo solución. Fuente: Elaboración propia.

El modelo propuesto en esta investigación consta de dos problemas de optimización: el

primero, busca establecer bloques de trabajo iniciales acorde al tiempo máximo de trabajo continuo, según las tablas de trabajo recibidas, y el segundo buscará combinar estos bloques creados para generar la menor cantidad de turnos de trabajo diario con un tiempo máximo de trabajo por día.

4.2.1. Notación matemática

Para el problema de Programación de Conductores CSP (Del inglés: *Crew Scheduling Problem*), que construye turnos a partir de porciones de tablas de servicios (En inglés: *Timetables*), mencionados en 4.2, se formula un modelo de programación lineal entero mixto, utilizando como parámetros para cada tabla, las horas de inicio y finalización de cada viaje, el lugar de salida y finalización de los recorridos, y la duración de estos. Adicionalmente, cada tabla presenta un conjunto de eventos ordenados en el tiempo, y cada evento t corresponde a un recorrido. Es importante denotar que en una sola tabla está contenido un solo servicio, y los viajes de la tabla son continuos, por lo que la hora y lugar de finalización del evento t es la misma hora y lugar de inicio del evento $t + 1$.

Para el primer problema, en donde se busca establecer los bloques iniciales, se formula un problema entero mixto (MIP), y para establecer los turnos combinando los bloques, se formula un problema de generación de columnas (CG), metodología recomendada para problemas de agendamiento como el Crew Scheduling Problem que surge en el sistema de transporte tratado.

Formulación matemática: Problema de partición de tablas en bloques

Conjuntos:

- T : Conjunto de tablas de trabajo
- E : Conjunto de eventos
- B : Conjunto de posibles bloques
- $B' \subset B$: Último posible bloque de trabajo
- I : Conjunto de estaciones intercambiadoras

Parámetros:

- $t0_{te}$: Hora de inicio del evento $e \in E$ de la tabla $t \in T$
- $inicio_{te}$: Lugar de inicio del evento $e \in E$ de la tabla $t \in T$
- tf_{te} : Hora de terminación del evento $e \in E$ de la tabla $t \in T$
- fin_{te} : Lugar de terminación del evento $e \in E$ de la tabla $t \in T$
- $duración_{te}$: Duración del evento $e \in E$ de la tabla $t \in T$
- $extra^b$: Minutos adicionales a las horas permitidas para un bloque
- MM : big M , valor muy grande para gestión de límites y restricciones en el modelo
- h^b : Máximo de horas en un bloque
- $balance$: Número de eventos de balance

Variables de decisión:

- y_{teb} : 1 si el evento $e \in E$ de la tabla $t \in T$ se asigna al bloque $b \in B$, 0 de lo contrario.

- x_{tb} : evento donde el bloque $b \in B$ termina en la tabla $t \in T$.
- nf_{tb} : 1 si el bloque $b \in B$ de la tabla $t \in T$ no es un bloque fantasma, 0 de lo contrario.

En este problema inicial, se establece el objetivo de maximizar el tamaño de los bloques acorde al tiempo máximo de trabajo continuo, buscando seleccionar el evento de terminación más tardío posible para cada uno de dichos bloques de la siguiente manera:

$$\max \sum_{t \in T} \sum_{b \in B} x_{tb} \quad (4.5)$$

s.a.

$$\sum_{b \in B} y_{teb} = 1 \quad \forall t \in T, e \in E, b \in B | t0_{te} > 0 \quad (4.6)$$

$$x_{tb} + 1 \leq x_{t(b+1)} \quad \forall t \in T, e \in E, b \in B | b \notin B' > 0 \quad (4.7)$$

$$x_{tb'} \leq \sum_{e \in E} \sum_{b \in B} y_{teb} + |B| - 1 \quad \forall t \in T, b' \in B' \quad (4.8)$$

$$t \cdot y_{teb} \leq x_{tb} \quad \forall t \in T, b \in B, e \in E \quad (4.9)$$

$$(1 - y_{te(b+1)}) \cdot MM + t \cdot y_{te(b+1)} \geq x_{tb} \quad \forall t \in T, b \in B, e \in E | b \notin B' \quad (4.10)$$

$$\sum_{e \in E} y_{teb} \cdot dur_{te} \leq nf_{tb} \cdot MM \quad \forall t \in T, b \in B \quad (4.11)$$

$$x_{t(b+1)} - x_{tb} \geq nf_{t(b+1)} \cdot balance \quad \forall t \in T, b \in B | b \notin B' \quad (4.12)$$

$$\sum_{e \in E} y_{leg} \cdot duracion_{le} \leq h^b + extra^b \quad \forall t \in T, b \in B \quad (4.13)$$

$$y_{t(e-1)b} + y_{teb} \geq 2 \cdot y_{t(e-1)b} \quad \forall t \in T, b \in B, e \in E | inicio_{te} \notin I \quad (4.14)$$

$$x_{tb} \in Z^t \quad (4.15)$$

$$y_{teb} \in \{0, 1\} \quad (4.16)$$

La meta de este problema inicial queda expresada en la función objetivo (4.5), la cual busca maximizar los eventos de corte seleccionados, abarcando la mayor cantidad de eventos posibles. De esta forma, se considera que todos los eventos de una tabla de trabajo se repartirán máximo en 4 bloques de trabajo, denotados bajo el conjunto G ; pues la máxima extensión de una tabla es alrededor de 20 horas de trabajo, donde se pueden particionar sin problema en cuatro bloques de 5 horas. Las restricciones (4.15) y (4.16) hacen referencia a la naturaleza de las variables. La restricción (4.6) asegura que un evento solo pueda asignarse a un posible bloque. Las restricciones (4.7) y (4.8) aseguran que los cortes de terminación de bloque sean continuos y no se traslapen, dando un espacio de tres eventos fantasma (sin servicios), para los posibles bloques que terminen como fantasmas. Un bloque fantasma (en referencia a la variable $n_{f_{tb}}$) se refiere a aquellos bloques que pueden quedar vacíos, pues al presentarse tablas de menos del tiempo máximo de trabajo continuo, por ejemplo, no se necesitará mas de un bloque para repartirla. En ese caso, los otros tres (3) bloques posibles se quedarían sin eventos y se les denominará fantasmas. La restricción (4.11) activa la variable $n_{f_{tb}}$ en los bloques que no quedan vacíos, y la restricción (4.12) asegura un balance en los bloques para que no queden con menos de siete (7) eventos en caso de que no sean fantasmas (este número de eventos es empírico y resultado de una consideración para el caso de las tablas de Integra S.A.). Las restricciones (4.9) y (4.10) relacionan las variables x_{tb} y y_{teb} para que los eventos seleccionados para cada bloque queden agrupados según los eventos de terminación de bloque. La restricción (4.13) asegura que los eventos agrupados en un bloque no sobrepasen el límite de bienestar de tiempo máximo de trabajo continuo, más un extra en minutos con el que actualmente la operación está considerada. Por último, la restricción (4.14) asegura que todos los bloques empiecen y finalicen en una estación intercambiadora por políticas de la compañía y capacidades logísticas.

Formulación matemática: Problema de generación de turnos de trabajo

Conjuntos

- T : Conjunto de tablas de trabajo
- E : Conjunto de eventos
- B : Conjunto de posibles bloques

Parámetros

- num : Número total de eventos
- $t0b_{tb}$: Hora de inicio del bloque $b \in B$ de la tabla $t \in T$
- tfb_{tb} : Hora de terminación del bloque $b \in B$ de la tabla $t \in T$
- $iniciob_{tb}$: Lugar de inicio del bloque $b \in B$ de la tabla $t \in T$
- $finb_{tb}$: Lugar de terminación del bloque $b \in B$ de la tabla $t \in T$
- $duracion_{tb}$: Duración del bloque $b \in B$ de la tabla $t \in T$
- a_{tbi} : 1 si el turno i atiende el evento $e \in E$ de la tabla $t \in T$. Matriz I con $i \leq num$
- $extra^t$: Minutos adicionales a las 10 horas permitidas para un turno.
- $descanso$: Descanso mínimo obligatorio entre bloques extremos e intermedio.
- $descanso^n$: Descanso nocturno obligatorio.
- h^t : Máximo de horas permitidas de un turno.

Variables de decisión

- w_i : 1 si se usa el turno i para satisfacer la demanda, o de lo contrario. Donde i pertenece a un rango indefinido.
- u_{tb} : 1 si el bloque $b \in B$ de la tabla $t \in T$ se selecciona en el turno, 0 de lo contrario.

- z_{te} : 1 si el evento $e \in E$ de la tabla $t \in T$ se selecciona en el turno, 0 de lo contrario.
- $DUAL_MP_{te}$: duales del problema maestro.
- $first_{tb}$: 1 si el bloque $b \in B$ de la tabla $t \in T$ es el primer bloque del turno, 0 de lo contrario.
- $middle_{tb}$: 1 si el bloque $b \in B$ de la tabla $t \in T$ es el bloque intermedio del turno, 0 de lo contrario
- $last_{tb}$: 1 si el bloque $b \in B$ de la tabla $t \in T$ es el último del turno, 0 de lo contrario.
- $ltime$: hora de inicio del turno.
- $utime$: hora de salida del turno.
- $bsolo$: 1 si el turno es un solo bloque, 0 de lo contrario.

Este problema es abordado con una generación de columnas con base en la solución del problema inicial de bloques, es decir, a partir de los bloques de trabajo formados, se realizará la combinación de estos en la menor cantidad de turnos posible. La solución del problema inicial se calcularían a través de las expresiones (4.17)-(4.21).

$$tob_{tb} = \min_{e \in E} 0_{te} \quad \forall t \in T, b \in B | y_{teb} > 0 \quad (4.17)$$

$$tfb_{tb} = \max_{e \in E} f_{te} \quad \forall t \in T, b \in B | y_{teb} > 0 \quad (4.18)$$

$$duracion_{tb} = \sum_{e \in E} y_{teb} * duracion_{te} \quad \forall t \in T, b \in B | y_{teb} > 0 \quad (4.19)$$

$$inicio_{tb} = inicio_{te} \quad \forall t \in T, b \in B, e \in E | tob_{tb} = to_{te} \quad (4.20)$$

$$finb_{tb} = fin_{te} \quad \forall t \in T, b \in B, e \in E | tfb_{tb} = tf_{te} \quad (4.21)$$

Entonces, se definen los mismos parámetros iniciales, ahora para cada posible bloque formado. Las expresiones (4.17) y (4.18) establecen los parámetros de horas de inicio y finalización

del bloque como los extremos de sus eventos. De manera análoga, con la ecuación (4.19), se asigna la duración del bloque como la suma de la duración de los eventos agrupados. Finalmente, con las ecuaciones (4.20) y (4.21) los lugares de inicio y finalización del bloque son definidos en donde coincidan las horas del evento correspondiente.

Así pues, el problema maestro se define a continuación:

$$\min \sum_i w_i c_i \quad (4.22)$$

s.a.

$$\sum_i a_{tei} * w_i = 1 \quad \forall t \in T, e \in E | t_{0te} > 0 \quad (4.23)$$

$$w_i \geq 0 \quad \forall i \in \text{Turnos} \quad (4.24)$$

$$w_i \leq 1 \quad \forall i \in \text{Turnos} \quad (4.25)$$

La variable w_i pasa a ser continua con las restricciones (4.24) y (4.25) como método de relajación en el algoritmo. El costo c_i en la función objetivo (4.22) es de uno (1) para todos los turnos, excepto para los turnos $i \leq num$, ya que la matriz a_{lti} consiste en una matriz identidad para inicializar el algoritmo, señalando que cada turno inicial atiende un solo evento o servicio. A estos turnos artificiales se les asigna un costo c_i de quince (valor empírico encontrado durante la fase de experimentación). La restricción (4.23) asegura que con los turnos generados, se atienda la totalidad de los eventos de las tablas. Ahora, este problema pasa a tener una cantidad de variables con explosión combinatoria, asociadas al turno que debería asignarse. Esto sucede porque las columnas en la matriz a_{lti} definen la solución de eventos programados en un turno, y esta

se encuentra inicializada con una matriz identidad. Como el problema maestro es próximo a las formulaciones de *Cutting Stock* y *Bin Packing*, se define un problema auxiliar similar para alimentar esta matriz con posibles turnos a asignar. Este problema auxiliar se asemeja al *Knapsack Problem*, y se generará usando principalmente las variables duales del grupo de restricciones (4.23):

$$DUAL_{MP_{te}} \forall t \in T, e \in E \quad (4.26)$$

Con estas variables duales definidas en cada iteración de la generación de columnas, se establece el problema auxiliar como la maximización de la oportunidad de mejora, esto quiere decir, que se buscará un turno que ayude a reducir el número total de turnos presente en la solución actual:

$$\max \sum_{t \in T} \sum_{e \in E} Z_{te} * DUAL_{MP_{te}} \quad (4.27)$$

s.a.

$$z_{te} = u_{tb}; \forall t \in T, e \in E, b \in B | y_{sol_{teb}} = 1 \quad (4.28)$$

$$\sum_{t \in T} \sum_{b \in B | t_0 b_{tb} > 0} u_{tb} \leq 3 \quad (4.29)$$

$$\sum_{t \in T} \sum_{b \in B | t_0 b_{tb} = 0} u_{tb} = 0 \quad (4.30)$$

$$\sum_{t \in T} \sum_{b \in B} first_{tb} = 1 \quad (4.31)$$

$$\sum_{t \in T} \sum_{b \in B} last_{tb} = 1 \quad (4.32)$$

$$\sum_{t \in T} \sum_{b \in B} middle_{tb} \leq 1 \quad (4.33)$$

$$first_{tb} \leq u_{tb}; \forall t \in T, b \in B \quad (4.34)$$

$$last_{tb} \leq u_{tb}; \forall t \in T, b \in B \quad (4.35)$$

$$middle_{tb} \leq u_{tb}; \forall t \in T, b \in B \quad (4.36)$$

$$\sum_{t \in T} \sum_{b \in B} first_{tb} + \sum_{t \in T} \sum_{b \in B} middle_{tb} + \sum_{t \in T} \sum_{b \in B} last_{tb} \geq \sum_{t \in T} \sum_{b \in B} u_{tb} \quad (4.37)$$

$$\sum_{t \in T} \sum_{b \in B} middle_{tb} \leq \frac{\sum_{t \in T} \sum_{b \in B} u_{tb} - 1}{2} \quad (4.38)$$

$$\sum_{t \in T} \sum_{b \in B} u_{tb} * duracion_{tb} \leq h^t + extra^t \quad (4.39)$$

$$ltime \leq t0b_{tb} * u_{tb} + (1 - u_{tb}) * MM; \forall t \in T, b \in B \quad (4.40)$$

$$ltime \geq t0b_{tb} * first_{tb}; \forall t \in T, b \in B \quad (4.41)$$

$$utime \leq tfb_{tb} * last_{tb} + (1 - last_{tb}) * MM; \forall t \in T, b \in B \quad (4.42)$$

$$utime \geq tfb_{tb} * u_{tb}; \forall t \in T, b \in B \quad (4.43)$$

$$24 * 60 - utime + ltime \geq descanso^n \quad (4.44)$$

$$bsolo \leq \frac{3 - \sum_{t \in T} \sum_{b \in B} u_{tb}}{2} \quad (4.45)$$

$$utime - ltime - \sum_{t \in T} \sum_{b \in B} u_{tb} * duracion_{tb} \geq (1 - bsolo) * descanso \quad (4.46)$$

$$\sum_{t \in T} \sum_{b \in B} first_{tb} * tfb_{tb} \leq \sum_{t \in T} \sum_{b \in B} middle_{tb} * t0b_{tb} + descanso + (1 - \sum_{t \in T} \sum_{b \in B} middle_{tb}) * MM \quad (4.47)$$

$$\sum_{t \in T} \sum_{b \in B} last_{tb} * t0b_{tb} \geq \sum_{t \in T} \sum_{b \in B} middle_{tb} * tfb_{tb} + descanso \quad (4.48)$$

Esta sección del modelo, presenta restricciones que buscan garantizar la factibilidad y la carga laboral de los turnos de trabajo que se generen. La restricción (4.28) controla las variables para

seleccionar eventos y bloques del turno, con respecto a la solución del problema de partición de tablas en bloques de trabajo resuelto inicialmente; de esta forma, esta restricción asocia todos los eventos que están contenidos en los bloques de trabajo posibles. Las restricciones (4.29) y (4.30) restringen a un máximo de tres (3) bloques para cada turno, y se asegura de que los posibles bloques considerados como fantasmas, no sean escogidos. Las restricciones (4.31), (4.32) y (4.33) garantizan la selección de bloques de inicio y de finalización; y un tercer posible bloque intermedio. Las restricciones (4.34), (4.35) y (4.36) relacionan las variables de asignación de bloques inicial, final e intermedio, seleccionados para conformar el turno de trabajo. Igualmente, la restricción (4.37) asegura que las variables de asignación de bloques estén activadas acorde a la cantidad de bloques asignados al turno. La restricción (4.38) garantiza que la variable que selecciona el bloque intermedio esté desactivada en caso de escoger menos de tres bloques. La restricción (4.39) restringe el tiempo de duración del turno de trabajo a lo definido por ley, más un extra de tiempo permitido. Las restricciones de la (4.40) a la (4.43) ajustan las variables de horas de entrada y de finalización en relación a los bloques de trabajo que se escojan. La restricción (4.44) asegura un tiempo de descanso nocturno entre la finalización del turno, hasta la entrada del siguiente día, con el fin de equilibrar las salidas tardías y las jornadas que inician en la madrugada. La restricción (4.45) controla una variable necesaria para definir turnos de un solo bloque si no es posible combinarlo con otros bloques de trabajo. La variable *bsolo* se utiliza en la restricción (4.46) para garantizar continuidad de los bloques, agregando un tiempo mínimo de descanso, siempre y cuando no se trate de un turno de un solo bloque. Adicionalmente, en caso de escoger tres bloques para un mismo turno, las restricciones (4.47) y (4.48) controlan las horas de inicio y finalización del bloque intermedio en relación a los bloques extremos, adicionando un tiempo de descanso entre los bloques que facilita el traslado en caso de que sea necesario. Este tiempo

controla también la generación de bloques de trabajo muy extensos a partir de la misma tabla de servicios. Finalmente, aunque se trabajen las restricciones con los bloques que se conectan para cada turno, la función objetivo del problema auxiliar (4.27), está relacionada con los eventos de las tablas de trabajo, considerando el costo de oportunidad de combinarlos en la solución del problema maestro. En cada iteración de la generación de columnas, se revisará que la función objetivo del problema auxiliar sea mayor que 1 para continuar. En cuanto sea igual a 1, significará que costo reducido de la función objetivo del maestro. Esto significa que no reducirá más los turnos de trabajo generados para agendar todos los eventos requeridos. La solución del problema auxiliar será guardada como una nueva columna de la matriz del problema maestro, así como también se considerará una nueva variable para considerar escoger ese nuevo turno generado en la medida que se avanza en el proceso:

$$zsol_{te} = a_{te}(\text{turno}) \quad (4.49)$$

Experimentos computacionales y análisis de resultados aplicados

Al tratarse de una metodología con enfoque matemático, se garantiza el cumplimiento de todas las restricciones tanto técnicas, planteadas por el centro de operaciones, bienestar laboral y regulaciones laborales vigentes, como de factibilidad, concernientes al encadenamiento de bloques de trabajo indexados en el tiempo. El modelo implementado por tanto, genera una combinación factible de turnos de trabajo en tiempos cortos de ejecución, usando instancias reales de un sistema de transporte masivo tipo BRT del Área Metropolitana Centro Occidente de Colombia, dicha instancias de prueba constan de hasta 3800 viajes requeridos. Estos resultados han sido comprobados con el coordinador de la operación, encontrando diferencias relevantes en

Tabla 4.2: Resultados obtenidos por el coordinador y el modelo propuesto para la instancia Mega060519

	Requerimiento	Número de servicios	Tipo de Turno				Total Conductores
			1 bloques	2 bloques	3 bloques	Mezclados	
Programación Manual del Coordinador	Rutas Alimentador (licencia C2)	3459	0	68	1	-	69
	Rutas Troncal (licencia C3)	892	0	46	3	4	53
Generación de Columnas	Rutas Alimentador (licencia C2)	3459	1	64	5	-	70
	Rutas Troncal (licencia C3)	892	0	41	8	0	49

comparación a la programación manual realizada.

El modelo fue programado usando C++ (Concert) y CPLEX 12.8. El tiempo de cómputo no sobrepasa los tres minutos para las instancias de tablas de trabajo de Alimentador, que suelen ser las tablas más extensas. Es importante resaltar que actualmente el proceso de programación de turnos de trabajo lleva un día completo de operación para los coordinadores.

El resultado principal se evaluó en las tablas más recientes de la operación del sistema, al comparar la programación manual realizada por el coordinador contra la metodología propuesta. En la Tabla 4.2 se registran los resultados del caso actual del sistema de transporte, programado del 6 de mayo de 2019, para la operación de un día hábil, la cual es la programación de mayor peso e importancia por la mayor cantidad de días de operación durante el año.

Para esta instancia se obtuvo un número menor de conductores. Ningún tipo de turno mezclado fue necesario. El surtido de turnos generados es bastante heterogéneo lo que permitiría una asignación más favorable, debido a los distintos requerimientos de la población de conductores.

4.2.2. Estudio para validación de metodologías

Las metodologías anteriormente presentadas, como se evidencia en los resultados, han mejorado las soluciones de los expertos del centro de control. Para validar estas aproximaciones,

se indagó en la literatura especializada casos que ajustaran, o que tengan un grado de generalidad suficiente, con el fin de obtener indicadores de rendimiento de los algoritmos propuestos en esta investigación. Es importante resaltar que los casos de prueba son difíciles de encontrar, porque suelen ser información reservada de empresas de transporte.

En [Beasley et al. \(1996\)](#), se proponen las instancias de prueba que se utilizarán para obtener indicadores de rendimiento. Estos casos de prueba se encuentran publicados en <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/cspinfo.html>. Como resaltan los autores, son casos que ajustan a diferentes contextos de generación de turnos. En el documento, presentan un algoritmo que deja en manos de una heurística el esfuerzo computacional para fijar el número de turnos resultantes, dado que los casos de prueba dejan abierta la cantidad de personal que se requiere para cada instancia. A continuación se presenta la formulación propuesta en [Beasley et al. \(1996\)](#).

Notación matemática [Beasley et al. \(1996\)](#)

El CSP se plantea como el problema de encontrar K caminos de vértices (tareas) disjuntos desde 0 hasta $N + 1$, de manera que:

- Todas las tareas pertenezcan a un camino.
- El tiempo de trabajo de cada camino no supere T , tiempo máximo que debe durar una jornada.
- El costo total de los caminos debe ser mínimo.

Definiendo:

$$x_{ij} = \begin{cases} 1, & \text{si el arco de transición } (i, j) \text{ es utilizado} \\ 0, & \text{de lo contrario} \end{cases} \quad (4.50)$$

El costo asociado a cada uno de los arcos se denomina c_{ij} , cuantificando el cambio de contexto entre las tareas i y j .

Formulación **Beasley et al. (1996)**

$$\min \sum_{i,j} c_{ij} x_{ij} \quad (4.51)$$

s.a.

$$\sum_k x_{jk} = \sum_i x_{ij} \quad j = 1, \dots, N \quad (4.52)$$

$$\sum_j x_{ij} = 1 \quad i = 1, \dots, N \quad (4.53)$$

$$\sum_j x_{0j} = K \quad (4.54)$$

$$\text{restricciones de límite de tiempo} \quad (4.55)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \quad (4.56)$$

Nótese que en la ecuación (4.55), los autores dejan abierta la forma en que se restringirá la duración de los turnos, es decir, que no excedan el tiempo máximo de trabajo T .

En esta formulación de CSP, la ecuación (4.52) especifica que el número de arcos saliendo

de una tarea es igual al número de arcos que entran a una tarea. La ecuación (4.53) especifica que cada tarea será asignada solamente a un camino. La ecuación (4.54) especifica que K tareas son seleccionadas para ser la primera tarea de un camino. En (4.56) se especifica la naturaleza binaria de la variable de decisión x_{ij} .

La propuesta de los autores asume que: (i) los costos de salida desde el depósito a la primera tarea no participan de la función objetivo, es decir, $c_{0,i} = 0, i \in \{1, \dots, N\}$. (ii) De la misma forma, los costos de regreso al depósito, después de atender la última tarea, no participan de la función objetivo, por lo tanto, $c_{i,N+1} = 0, i \in \{1, \dots, N\}$. Finalmente, el tiempo b_i , correspondiente al tiempo de servicio en una tarea i , para el caso de los depósitos, $b_0 = b_{N+1} = 0$.

El modelo presentado entonces, prioriza la disminución del costo de los turnos recibidos de la heurística descrita en el estudio, pero no incluye la disminución de los mismos. Como se puede observar, la restricción (4.54) encargada de este aspecto, es una igualdad, fijando el número de turnos.

Formulación propuesta

Dado que la minimización del número de turnos no está integrada en el modelo con el cual se resuelven las instancias de la literatura, se propone un modelo que además de la minimización de los costos, incluya la minimización del número de turnos, como se requiere en el caso real. Integrando el manejo de las cadenas propuesto en [Mesquita and Paixão \(1992\)](#), y utilizando restricciones de flujo en cadenas de nodos como sugiere [Borcinova \(2017\)](#), se propone el siguiente nuevo modelo.

En esta formulación se heredan los elementos de la notación de [Beasley et al. \(1996\)](#) con algunas especificaciones adicionales:

Notación y costos

- V : conjunto de vértices que representan los bloques de trabajo o servicios que se van a programar, junto con el depósito virtual 0, donde $V = \{0, \dots, N\}$.
- A : conjunto de arcos o transiciones permitidas (i, j) entre bloques de trabajo, cumpliendo restricciones técnicas y ventanas de tiempo de los servicios, donde $i \in V \wedge j \in V \wedge i \neq j$.
- A' : conjunto de arcos o transiciones permitidas (i, j) entre bloques de trabajo, cumpliendo restricciones técnicas y ventanas de tiempo de los servicios, sin tener en cuenta el depósito virtual, donde $i \in V \setminus \{0\} \wedge j \in V \setminus \{0\} \wedge i \neq j$.
- G : grafo incompleto que representa todas las posibilidades de programación de itinerarios constituido como $G = (V, A)$.
- c_{ij} : costo de transición entre la tarea i y la tarea j .
- Cuando $(i, j) \notin A'$, $c_{ij} = +\infty$. Los arcos a los cuales se le asociará este costo, son aquellos que no cumplen restricciones técnicas, son bucles en el grafo G , o presentan traslapes en las ventanas de tiempo respectivas de i y j . Nótese que en las ventanas de tiempo de los bloques de trabajo, el tiempo de servicio, inicio y finalización, se encuentran totalmente ajustados.
- r : costo de transición más alto entre servicios, es decir, $\max(c_{ij}), (i, j) \in A'$
- $c_{0i} = 2 \cdot r; j \in \{1, \dots, N\}$. Costo reactivo según la instancia, para penalizar la apertura de turnos y por ende la ocupación de más personal.
- s_i : hora de inicio de la tarea i .
- f_i : hora de finalización de la tarea i .
- b_i : tiempo de servicio o duración de la tarea i .

- y_i : tiempo transcurrido hasta la tarea i en el k -ésimo turno o cadena.

Modelo matemático propuesto

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (4.57)$$

s.a.

$$\sum_{(i,j) \in A'} x_{ij} \leq 1 \quad i \in V \setminus \{0\} \quad (4.58)$$

$$\sum_{(i,j) \in A} x_{ij} = 1 \quad j \in V \setminus \{0\} \quad (4.59)$$

$$\sum_{j \in V \setminus \{0\}} x_{0j} \leq K \quad (4.60)$$

$$y_i + (b_j + (s_j - f_i))x_{ij} - T(1 - x_{ij}) \leq y_j \quad (i, j) \in A' \quad (4.61)$$

$$b_i \leq y_i \leq T \quad i \in V \setminus \{0\} \quad (4.62)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (4.63)$$

El objetivo, formalizado en la ecuación (4.57), consiste en minimizar el número de turnos y el costo de cada uno de ellos. El conjunto de ecuaciones (4.58), restringe a 1 el número de servicios que se puede realizar después de hacer un servicio i , o bien, regresar al depósito para finalizar la cadena de servicios. El conjunto (4.59), obliga realizar una tarea después de salir del depósito, o posterior a la atención de una tarea. Esta última restricción debe ser de igualdad para asegurar la asignación de todos los bloques de trabajo a un turno, es decir, garantizar que todas las tareas o servicios serán atendidos. La ecuación (4.60) limita el número de turnos que se pueden programar, dependiendo del recurso humano cuantificado en K . En las instancias de prueba, este valor es

ilimitado, sin embargo en la práctica hay una cantidad específica de personal disponible. Para resolver los casos de [Beasley et al. \(1996\)](#), en este trabajo se propone una heurística para calcular el valor de K . Finalmente, las restricciones (4.61 y 4.62), de manera análoga a la forma como la carga transportada a través de rutas abiertas CVRP es controlada en [Borcinova \(2017\)](#), el modelo propuesto restringe la duración de los turnos con el encadenamiento facilitado por la variable y_i , evitando que la carga laboral sea mayor a T , valor que suele ser de 8 horas o 480 minutos.

La presente formulación, ofrece una versión de [Beasley et al. \(1996\)](#) completa, con un manejo de encadenamiento de bloques de trabajo adecuado como es el de [Borcinova \(2017\)](#), y especialmente, fortalece al modelo original al minimizar tanto el número de operadores requeridos, como el costo de los cambios de contexto entre cada uno de los bloques de trabajo. Naturalmente, manejar estos dos objetivos de manera simultánea dentro del modelo, puede generar un incremento de la dificultad para solucionarlo. Los experimentos y la discusión correspondiente se presentan más adelante en este trabajo.

4.2.3. Implementación del estudio comparativo

Proceder a la implementación del modelo propuesto, desencadenó un estudio interesante de *solvers* y algoritmos adecuados para obtener los óptimos tanto en costos como en número de operadores en el mejor de los casos, o bien, resultados de buena calidad para proceder con el estudio comparativo requerido para la metodología que resuelve exitosamente el caso real de programación, presentada en [4.2](#).

Como resultado de la exploración mencionada, además de utilizar el solver CPLEX® 12.9 para resolver el modelo propuesto, se encontró en la literatura especializada un solver

recientemente publicado, orientado a problemas de enrutamiento y sus variantes. Los solvers de propósito general como CPLEX suelen requerir la incorporación de mecanismos específicos para lograr convergencia, sin embargo, en [Pessoa et al. \(2020\)](#), los autores ofrecen un framework similar al que ofrece CPLEX como solver de propósito general, y Concert como librería para realizar el modelado, sin embargo, aportan un nivel de abstracción mayor, construyendo un solver que tiene CPLEX 12.9 o 12.10 embebido, junto con librerías de alto rendimiento y algoritmos especializados para problemas de enrutamiento y afines que los mismos autores han publicado y desarrollado previamente, demostrando la efectividad de estos enfoques. En este caso, la interfaz está construída en el lenguaje de programación Julia <https://docs.julialang.org/en/v1/>, que de la misma forma que Concert, siguiendo la generalización de encadenamiento de arcos de [Pessoa et al. \(2020\)](#), nos permite o utilizar las plantillas para problemas clásicos, o reformular un problema específico, como es el CSP, uno de los componentes principales de este trabajo.

Los autores denominan este framework como *VRPSolver*, presentados en el siguiente sitio web y su documentación:

- **Sitio Web:** <https://vrpsolver.math.u-bordeaux.fr/>
- **Documentación:** <https://vrpsolver.math.u-bordeaux.fr/doc/index.html>

Modelamiento en el *VRPSolver*

Los autores en [Pessoa et al. \(2020\)](#), plantean que adaptar y reimplementar las actuales versiones de los algoritmos de *Branch-and-Cut-and-Price* (BCP), las cuales se han sofisticado notablemente durante los últimos años, puede ser una tarea compleja y extensa. Por esta razón, proponen un solver BCP para un modelo genérico, el cual está en capacidad de resolver un conjunto

amplio de variaciones de problemas de enrutamiento de vehículos (VRP, por sus siglas en inglés Vehicle Routing Problems).

Un avance notable en el BCP está publicado en [Pecin et al. \(2017b\)](#), versión que ha sido incorporada en este framework del VRPSolver, y que permite abordar con enfoque exacto, problemas con una cantidad de nodos (clientes, bloques de trabajo, tareas, etc.) mayor. Subsecuentemente, en [Pecin et al. \(2017a\)](#), se ha mostrado la efectividad del BCP mencionado en problemas de enrutamiento de vehículos considerando ventanas de tiempo (VRPTW, por sus siglas en inglés Vehicle Routing Problem with Time Windows). Este último, indica que esta versión del BCP es efectiva cuando existen ventanas de tiempo en los nodos, situación común con el CSP.

Por otro lado, los autores en [Pessoa et al. \(2020\)](#), retoman el enfoque de [Desaulniers et al. \(1998\)](#) que busca generalizar la representación de problemas de enrutamiento y programación o agendamiento a través del problema de RCSP (por sus siglas en inglés *Resource Constrained Shortest Path*), cuya formulación clásica se encuentra en [Beasley and Christofides \(1989\)](#). El algoritmo BP aplicado en [Desaulniers et al. \(1998\)](#), siendo una versión mucho menos sofisticada que el BCP del VRPSolver, ya mostraba ser efectivo para problemas donde las ventanas de tiempo son muy ajustadas, situación que favorece la naturaleza del problema de este estudio, donde los bloques de trabajo tienen un tiempo de servicio completamente ajustado, es decir, idéntico al tamaño de la ventana de tiempo del mismo. Adicionalmente, el BCP incorpora elementos del reciente trabajo [Sadykov et al. \(2020\)](#) para problemas que manejan ventanas de tiempo, haciendo de este framework adecuado para abordar por lo menos una versión genérica del CSP.

Así las cosas, la definición formal del modelo genérico tipo RCSP sobre el cual trabaja el BCP implementado en el VRPSolver se presenta a continuación. Al utilizar este modelo, la idea

es mapear las variables introducidas por los caminos que consumen el recurso restringido a una función objetivo definida por el usuario y restricciones adicionales si es el caso en la formulación maestra o principal.

Definición del Grafo para los Subproblemas RCSP: Sea K el conjunto de subproblemas. Se definen los grafos dirigidos $G^k = (V^k, A^k), k \in K$. Sea $V = \cup_{k \in K} V^k$ y $A = \cup_{k \in K} A^k$. Los grafos no son necesariamente simples y podrían incluso tener loops. Vértices y arcos en todos los grafos son diferentes. Cada grafo tiene vértices fuente y sumidero, es decir v_{fuente}^k y $v_{sumidero}^k$ respectivamente. La fuente y el sumidero pueden ser vértices diferentes, o incluso ser el mismo vértice. Se define el conjunto R^k de recursos. Para cada $r \in R^k$ y $a \in A^k, q_{a,r} \in \mathbb{R}$ es el consumo del recurso r en el arco a . Se establecen intervalos finitos de consumo de recurso $[l_{a,r}, u_{a,r}]$, para cada $r \in R^K$ y $a \in A^k$. Dado que en la mayoría de aplicaciones dichos intervalos se definen naturalmente en los vértices, los autores, para dar mayor claridad, definen los intervalos como $[l_{v,r}, u_{v,r}], v \in V^k$, expresando que $[l_{a,r}, u_{a,r}] = [l_{v,r}, u_{v,r}]$, para cada arco $a \in \delta^-(v)$, es decir, entrando al vértice v . Los recursos sin consumos negativos son denominados *monótonos*, y *no-monótonos* de lo contrario. El conjunto R^k está dividido en *recursos principales* R_M^k y *recursos secundarios* R_N^k . Los recursos principales deberían ser monótonos, mientras que los secundarios podrían ser monótonos y no-monótonos. Esta diferenciación es relevante principalmente en la implementación de la solución, es decir, el VRPSolver. Para cada ciclo en G^k , el consumo de la red de al menos un recurso debería ser estrictamente positivo. Por lo tanto, a menos que G^k sea acíclico, la existencia de por lo menos un recurso es obligatoria. Finalmente, los recursos son también clasificados como *disponibles* y *no-disponibles*. En el primer caso, es posible liberar recursos si esto es necesario para satisfacer los intervalos acumulados de consumo de recursos. Los recursos principales son siempre de

tipo disponible. Por defecto, los recursos secundarios son también asumidos como disponibles.

La existencia de recursos no-disponible debe ser indicada de manera explícita. Un camino con

restricción de recursos $p = (v_{fuente}^k = v_0, a_1, v_1, \dots, a_{n-1}, v_{n-1}, a_n, v_n = v_{sumidero}^k)$ en el grafo G^K

debe tener $n \geq 1$ arcos, $v_j \neq v_{fuente}^k$ y $v_j \neq v_{sumidero}^k$, $1 \leq j \leq n - 1$ y es factible si:

- Para cada $r \in R^k$ que está disponible, el consumo acumulado del recurso $S_{j,r}$ en la visita j , $0 \leq j \leq n$, donde $S_{0,r} = 0$ y $S_{j,r} = \max\{l_{a_j,r}, S_{j-1,r} + q_{a_j,r}\}$, no excede $l_{a_j,r}$.
- Para cada $r \in R^k$ que es no-disponible, el consumo acumulado del recurso $S_{j,r}$ en la visita j , $0 \leq j \leq n$, donde $S_{0,r} = 0$ y $S_{j,r} = S_{j-1,r} + q_{a_j,r}$, se encuentra dentro del intervalo $[l_{a_j,r}, u_{a_j,r}]$.

Nótese que algunos caminos factibles podrían no ser elementales, algunos vértices o arcos podrían ser visitados más de una vez. Para cada $k \in K$, denotando con P^k el conjunto de todos los caminos con recurso restringido factibles en G^k . Cada conjunto P^k es finito, bien sea porque G^k es acíclico, o porque los recursos limitan el número de veces que cada vértice o arco puede ser visitado. Se define $P = \cup_{k \in K} P^k$. Como los vértices y los arcos en diferentes grafos no son los mismo, los caminos en diferentes grafos tampoco son iguales.

Formulación Maestra: Para todo $a \in A$ y $p \in P$, teniendo h_a^p indicando cuántas veces el arco a aparece en el camino p . Dado $c \in \mathbb{R}^{n_1}$, $f \in \mathbb{R}^{n_2}$, $\alpha \in \mathbb{R}^{m \times n_1}$, $\beta \in \mathbb{R}^{m \times n_2}$, y $d \in \mathbb{R}^m$, el problema es formulado de la siguiente manera:

$$\min \sum_{j=1}^{n_1} c_j x_j + \sum_{s=1}^{n_2} f_s y_s \quad (4.64)$$

s.a.

$$\sum_{j=1}^{n_1} \alpha_{ij} x_j + \sum_{s=1}^{n_2} \beta_{is} y_s \geq d_i \quad i = 1, \dots, m \quad (4.65)$$

$$x_j = \sum_{k \in K} \sum_{p \in P^k} \left(\sum_{a \in M(x_j)} h_a^p \right) \lambda_p \quad j = 1, \dots, n_1 \quad (4.66)$$

$$L^k \leq \sum_{p \in P^k} \lambda_p \leq U^k \quad k \in K \quad (4.67)$$

$$\lambda_p \in \mathbb{Z}_+ \quad p \in P \quad (4.68)$$

$$x_j \in \mathbb{Z}, y_s \in \mathbb{Z} \quad j = 1, \dots, \bar{n}_1; s = 1, \dots, \bar{n}_2 \quad (4.69)$$

Donde x_j , $1 \leq j \leq n_1$, y_s , $1 \leq s \leq n_2$, y λ_p , $p \in P$ son variables. Las primeras \bar{n}_1 x variables y las primeras \bar{n}_2 y variables están definidas como enteras; todas las variables λ son enteras no negativas. Las ecuaciones (4.64) define una función objetivo lineal, c y f son vectores de costos. Las desigualdades (4.65) definen m restricciones lineales sobre esas variables, α y β son matrices de coeficientes y d el vector de la derecha. Las restricciones (4.65) podrían contener una cantidad exponencial de familias de cortes, dado que son generados por su separación. Sin embargo, por simplicidad, se asume que la totalidad de las m restricciones son explícitamente definidas. Para cada variable x_j , $1 \leq j \leq n_1$, $M(x_j) \subseteq A$ se define el mapeo a un subconjunto no vacío de arcos. Es importante anotar que no es necesario que los mapeos sean disjuntos, el mismo arco puede ser mapeado a más de una variable x_j . Se define $M^{-1}(a)$, como $\{j \mid a \in M(x_j)\}$. Dado que no todos los arcos necesitan pertenecer a un mapeo, algunos conjuntos M^{-1} podrían ser vacíos. La relación

entre las variables x y λ es dada en (4.66). Para cada $k \in K$, L^k y U^k son dados límites inferiores y superiores en el número de caminos desde G^k en una solución.

Formulación VRPTW (PDPTW) bajo el esquema propuesto: Los datos de entrada para la formulación serían el grafo dirigido $G = (V, A)$, donde $V = \{0\} \cup P \cup D$, $P = \{1, \dots, n\}$ es el conjunto de vértices de recolección y $D = \{n+1, \dots, 2n\}$ el conjunto de entregas correspondientes (una recogida en i corresponde a una entrega en $i+n$); Q es la capacidad de los vehículos; c_a el costo de viaje y t_a el tiempo (incluido el tiempo de servicio) para $a \in A$; d_v las demandas positivas, $v \in P$ ($d_v = -d_{v-n}$, $v \in D$); y las ventanas de tiempo $[l_v, u_v]$, $v \in V$.

El objetivo es encontrar un conjunto de rutas de mínimo costo, empezando y terminando en el depósito, realizando todas las recogidas y entregas dentro de las ventanas de tiempo (se permiten esperas) tal que, a través de la ruta, la demanda total que ya ha sido recolectada pero no ha sido entregada aún, no exceda la capacidad del vehículo.

Modelando el CSP genérico en el VRPSolver: En este trabajo el modelo presentado en 4.2.3, es ajustado para el problema de programación de conductores genérico GCSP (Del inglés: *Generig Crew Scheduling Problem*) presentado en Beasley et al. (1996) y reformulado de manera completa en la sección 4.2.2 de este trabajo. Siguiendo la estructura del modelo maestro en 4.2.3 tendríamos las siguientes especificaciones para la formulación:

- La formulación está orientada a aristas conformadas por las transiciones factibles entre servicios especificadas en cada instancia, que en este caso corresponde al conjunto A . De la misma forma, el costo c_a para $a \in A$ es el costo interbloque establecido en la librería presentada por Beasley et al. (1996).

- La capacidad del vehículo Q se establece ilimitada, porque el modelo está orientado a la duración de los itinerarios o de los *Shortest Path* genéricos, por lo tanto $Q = \infty$.
- El tiempo total de servicio T permitido, corresponde al máximo de una jornada laboral, que son 8 horas para [Beasley et al. \(1996\)](#), como suele establecerse en la mayoría de las regulaciones, por lo tanto, $T = 480$.
- Cada arista $a \in A$ está compuesta por dos vértices v_1 y $v_2 \in V$. Todos los servicios $v \in V$ tienen asociada una ventana de tiempo $[l_v, u_v]$, cuyo tiempo de servicio v_{ts} se encuentra completamente ajustado, es decir, $v_{ts} = u_v - l_v$. Por lo tanto tenemos 3 casos de consumo de tiempo asociado para cada arista: (i) cuando $v_1 = 0 \wedge v_2 \neq 0$, el cambio de contexto desde el depósito hasta atender el primer servicio no genera consumo de tiempo (desplazamiento para iniciar la jornada laboral), por lo tanto $t_a = v_2 v_{ts}$; (ii) cuando $v_1 \neq 0 \wedge v_2 = 0$ que corresponde a un retorno al depósito o simplemente completar la jornada que no está asociado a ninguna tarea del turno, por lo tanto el consumo de tiempo es $t_a = 0$; y (iii) $v_1 \neq 0 \wedge v_2 \neq 0$, correspondiente a la transición entre bloques de trabajo, donde el consumo de tiempo asociado a la arista es $t_a = l_{v_2} - u_{v_1} + v_2 v_{ts}$.
- La función objetivo se planteó de 3 formas diferentes para los experimentos diseñados en este estudio, presentados en la sección 5. La de mayor interés, consiste en la minimización de los c_a de las aristas seleccionadas, penalizando el inicio de cada de la misma forma que se plantea en la formulación propuesta 4.2.2 dada la importancia que tiene en este estudio la minimización de utilización de personal (itinerarios). Las otras dos variantes corresponden a la minimización de los costos interbloque únicamente para realizar un comparativo directo con [Beasley et al. \(1996\)](#), donde el número de conductores se fija en el modelo a partir de

una heurística. La tercera forma buscó observar el comportamiento del modelo al minimizar el número de aristas a , donde $v1 = 0$, es decir, aristas salientes del depósito.

Una vez descritos de manera formal los modelos involucrados en la implementación del estudio comparativo, en la Figura 4.14 se muestran las capas que componen el estudio.

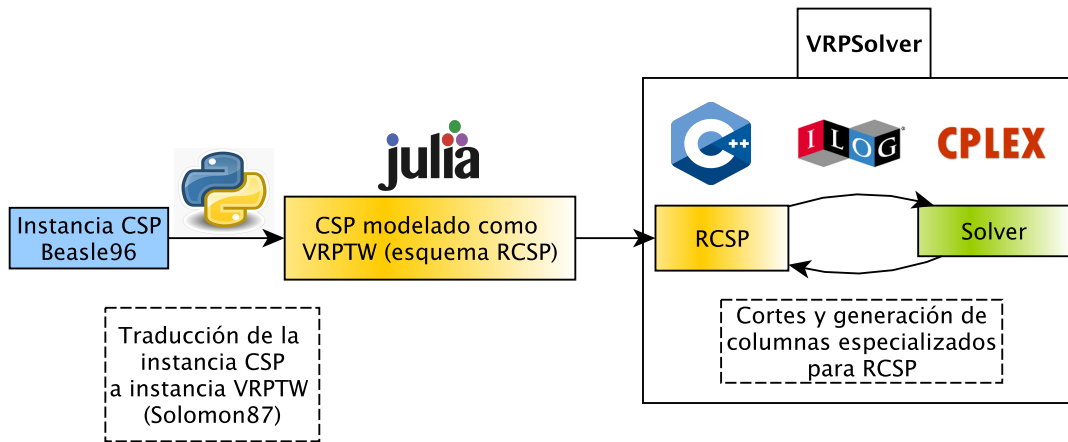


Figura 4.14: Componentes del estudio comparativo. Fuente: Elaboración propia.

Comúnmente, la implementación de enfoques matemáticos, suele ser el modelamiento directo, utilizando algún lenguaje de programación, usualmente C++ para alto rendimiento, a través de una librería que asiste la construcción de los componentes del modelo en formato estándar, con el fin de entregárselo a un solver, en este caso CPLEX. Dicho solver, como se ha discutido previamente en este capítulo, es de propósito general, y requiere la implementación de mecanismos especializados para problemas de asignación de tareas o generación de rutas; en general, problemas que caben en el marco RCSP desarrollado por Pessoa et al. (2020). La propuesta que se presenta en esta sección, habiendo iniciado como un experimento necesario para encontrar los óptimos y tiempos computacionales de la librería para el problema de programación de conductores genérico GCSP (Por sus siglas en inglés *Generic Crew Scheduling Problem*), acaba

aportando las mejores soluciones conocidas para el repositorio, y un marco de trabajo que resuelve eficientemente problemas de programación de conductores.

Así las cosas, dicho marco de trabajo presentado en la Figura 4.14, ofrece un enfoque de trabajo para los problemas de esta naturaleza, el cual empieza a observarse en la literatura especializada, donde se utilizan lenguajes de programación flexibles que agilizan la implementación del proceso de carga de las instancias, formateo de los parámetros y la generación del modelo que se entregará a un solver. Específicamente, como se muestra en el gráfico, se utiliza Python para convertir las instancias CSP de [Beasley et al. \(1996\)](#) al formato de las instancias VRPTW (Por sus siglas en inglés *Vehicle Routing Problem with Time Windows*) en el formato de [Solomon \(1987\)](#). Luego el VRPSolver cuenta con una interfaz implementada en el lenguaje Julia, para la carga de la información de los parámetros del modelo, la construcción del mismo, y el llamado al núcleo de este paquete de software, que presenta la implementación del RCSP en C++ y Concert, junto con mecanismos especializados de cortes y generación de columnas a través de callbacks, y la librería de alto rendimiento *boost* escrita en C++, que implementa recorridos optimizados sobre grafos.

Capítulo 5

Análisis resultados

Los estudios computacionales de esta investigación están centrados en la evolución de la metodología en el frente del problema de programación de conductores, dado que en [Moreno et al. \(2019\)](#) y [Marín-Moreno \(2019\)](#) se presentan estudios computacionales detallados de la metodología de programación de flota integrada en este trabajo.

Todas las implementaciones que se realizaron, han sido bajo diferentes versiones de la distribución Ubuntu (16.04 y 20.04), es decir, sistemas operativos Linux de 64 bits, adecuados para este tipo de algoritmos de alto rendimiento.

En las siguientes secciones se presenta un análisis del comportamiento de los diferentes enfoques previamente presentados en el capítulo [4](#).

5.1. Enfoque heurístico

En la sección [4.1](#) se presenta la implementación realizada para resolver el problema real de CSP en el sistema de transporte masivo tipo BRT estudiado. Este experimento deja un algoritmo

que utiliza eficientemente el personal, disminuyendo el número de operadores e integrando restricciones algorítmicamente que pueden generar una dificultad importante en enfoques de solución matemáticos, como el control de traslape temporal de los bloques, los tiempos permitidos de descanso, asociar las transiciones entre bloques de trabajo a restricciones de locación específica, intentar que la mayoría de los turnos de trabajo estén constituídos con dos bloques y en el peor de los casos tres bloques, y que el tiempo interbloque sea suficiente para descansar, pero no muy amplio para que un porcentaje del día mayor el conductor pueda realizar otras actividades que sumen en el bienestar laboral. El resultado obtenido para la operación se presenta en la Figura 5.1. En esta, se presenta en las filas (eje vertical) un itinerario de trabajo, donde cada una de las líneas continuas de color verde (vehículos articulados) o color amarillo (vehículos de alimentación), corresponden a bloques de trabajo de máximo cinco horas, conformando turnos para los operadores idealmente entre 8 y 10 horas. En cada columna (eje horizontal) se indexan las horas del día, recordando que el horizonte de tiempo de los problemas, son las horas de una sola jornada. Las configuraciones ideales, conjugando el bienestar laboral, y la ocupación del personal, son aquellas que presentan dos bloques tan cercanos como sea posible a las 5 horas, y turnos entre 8 y 10 horas de trabajo, con un tiempo suficiente, mas no exagerado, de descanso entre dichos bloques continuos. Equilibrar todos estos aspectos es complicado, especialmente en un algoritmo constructivo, sin embargo, en la Figura 5.1, reportando disminución del número de conductores necesarios para satisfacer los servicios que deben ser atendidos, obtiene turnos deseables en dos tercios de los que produce la metodología. Los turnos que son menos atractivos, son aquellos que se encuentran en las primeras filas de la Figura 5.1, porque a pesar de que presentan dos bloques de trabajo bien diferenciados y continuos, hay mucho espacio entre estos, situación que va en contra del bienestar laboral del conductor, al quedar todo el día en función del segundo

bloque de trabajo. La solución obtenida por el algoritmo constructivo, naturalmente está sujeta a mejoras, porque además de producir algunos turnos de un solo bloque de trabajo, presentan unas pocas horas de ocupación. Esto último no es conveniente, dado que la vinculación de los empleados no es por horas, si no que se trata de una vinculación a tiempo completo, y esta fuerza laboral puede verse desperdiciada, al tener operadores que estarían simplemente disponibles para contingencias, o realizando otras tareas diferentes a la operación de los vehículos, pero generando gastos al operador como si estuvieran ejecutando turnos de operación.

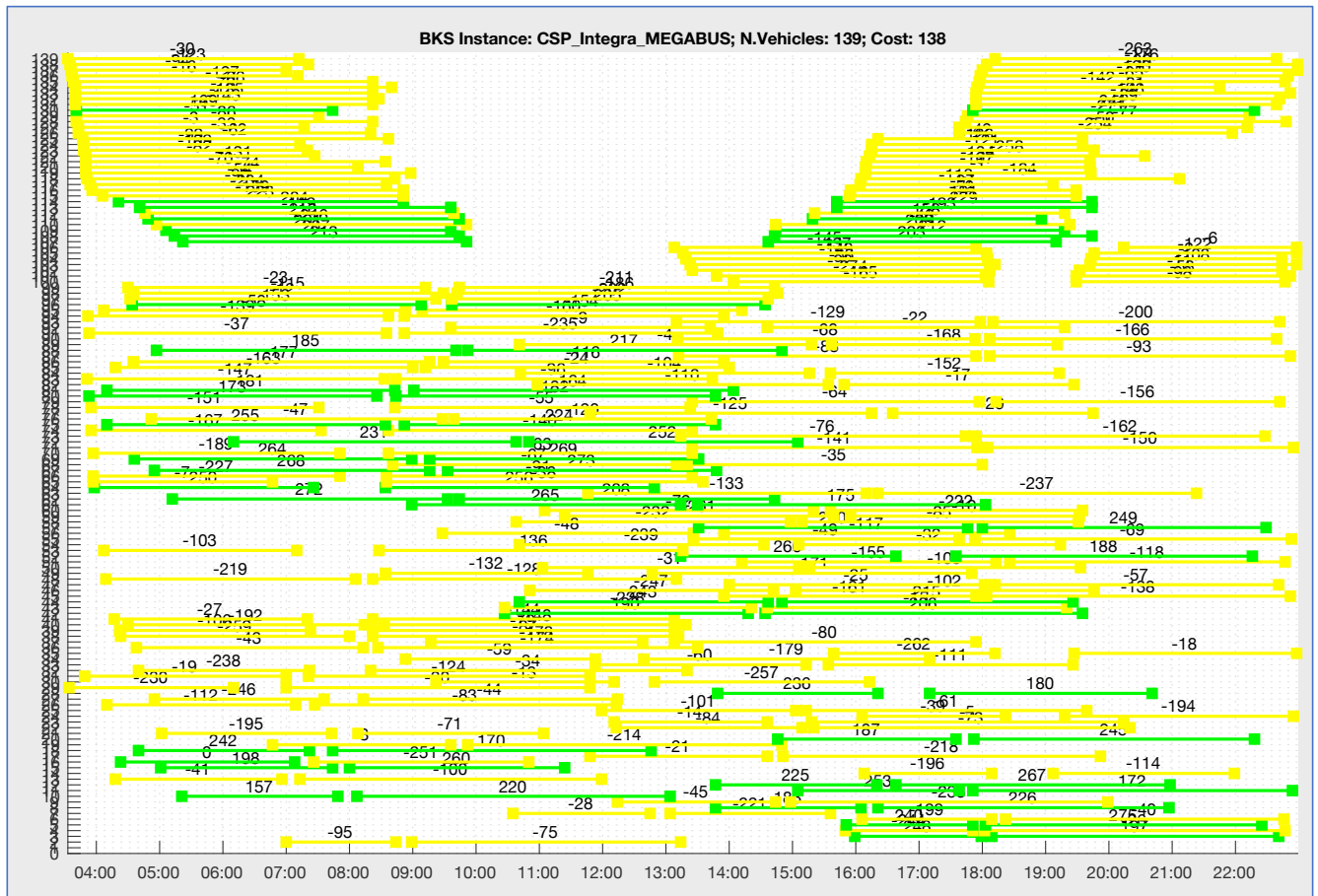


Figura 5.1: Resultado turnos de trabajo caso Megabús 2019: Amarillo Alimentación, Verde Articulados. Fuente: Elaboración propia.

Una proporción importante entonces, cumple con la mayoría de restricciones sugeridas por los expertos en el área de control, sin embargo, no logra que la totalidad de los turnos cumplan con

los objetivos de bienestar laboral (tiempos interbloque muy largos en unos pocos casos), que si bien son opcionales para cumplir de manera estricta con la operación, son factores que al largo plazo son relevantes para disminuir la rotación de personal, lo que conlleva a sobre costos de capacitación, y dificultades en la operación por fuga de personal experto. Los tiempos computacionales obtenidos son los más adecuados cuando se requiera realizar una programación muy rápida y cumpliendo con las restricciones fuertes como se indica en la sección 4.1.3, donde se consignan resultados de buena calidad con tiempos de respuesta muy cortos.

5.2. Enfoques matemáticos

Los enfoques matemáticos fueron la siguiente fase en la investigación por dos razones principalmente: (i) las limitaciones de la heurística para alcanzar soluciones equilibradas entre eficiencia de la asignación de personal y su bienestar laboral; y (ii) este tipo de problemas de agendamiento no suelen tener los mejores resultados a través de enfoques metaheurísticos, dado que la cantidad de restricciones, incluso mayor que en el caso del agendamiento de vehículos, generan una pérdida importante del esfuerzo computacional de un enfoque netamente algorítmico, a causa del alto grado de infactibilidad, por lo tanto, se plantea entre otras estrategias exactas, un enfoque clásico como es la generación de columnas para construir los turnos de trabajo de los conductores.

En la sección 4.2.1, se presenta la metodología desarrollada para el problema real de operación en el sistema de transporte objeto de estudio. En los comparativos siguientes, esta forma de abordar el problema será referida en el documento como **GC** (Generación de Columnas), la cual alcanza de manera exitosa la disminución del número de conductores, teniendo además en cuenta los criterios

de equilibrio de carga laboral y descansos adecuados. La implementación del **GC** presentó muy buenos resultados en la programación real del sistema BRT, sin embargo, fue necesario realizar un estudio comparativo para establecer la efectividad y tiempo de cómputo de esta propuesta.

Como se ha mencionado previamente en el documento, los detalles de la planeación operativa de casos reales, suelen ser información reservada de las empresas del sector transporte y las restricciones de cada problema difieren mucho; por tal razón, en la literatura especializada se opta por trabajar con instancias genéricas del problema como ofrecen los autores de [Beasley et al. \(1996\)](#). Así como sucede en esta investigación, los autores de [Derigs and Schäfer \(2014\)](#), para trabajar las variantes realistas de su interés, tomaron la librería de [Beasley et al. \(1996\)](#) para probar sus dos propuestas de solución, y determinar su efectividad primero en este problema de agendamiento de conductores genérico GCSP (Del inglés: *Generic Crew Scheduling Problem*), y luego involucrando otros aspectos como la minimización de tiempos ociosos entre bloques de trabajo, y un horizonte de tiempo hasta el día siguiente para la constitución de los turnos.

Dado que [Beasley et al. \(1996\)](#) presenta una formulación parcial, en esta investigación se propone un modelo matemático completo para el GCSP. Dicho modelo es implementado y presentado en los resultados como **Modelo Completo**. Adicionalmente, los autores de [Beasley et al. \(1996\)](#) dejan en manos de una heurística el número de conductores necesarios para atender la totalidad de los servicios, complementado con un barrido de dos turnos hacia arriba y dos turnos hacia abajo a partir del número de operadores obtenido, antes de la constitución de los itinerarios al resolver numéricamente el modelo que proponen. Es natural que los autores hayan hecho este modelamiento parcial y fijado dicho valor, dada la limitada capacidad de cómputo del momento en el que se realizó el estudio, liberando el proceso de solución del modelo, de la carga computacional

de minimizar el número de turnos de trabajo, concentrándose en la minimización de los costos interbloque.

Naturalmente, el costo de la formulación, a pesar de realizar el experimento en CPLEX 12.8 y disponiendo de hardware con mayor capacidad, el solver de propósito general tiene dificultades resolviendo el modelo, incluso utilizando una versión ligera de la heurística de construcción de turnos (4.1), cumpliendo solamente lo requerido por el GCSP para dar una solución inicial y factible al modelo. Al reportar GAPS entre el *Lower Bound* y el *Upper Bound*, aún no era posible garantizar que el número de conductores y costos reportados por el modelo eran óptimos, necesarios para establecer el rendimiento del GC (4.2.3). En la exploración de la literatura especializada entonces, para completar el estudio computacional de obtener una base comparativa, se encontró la propuesta de los autores Pessoa et al. (2020), planteando un solver especializado en problemas que se puedan generalizar en un RCSP como se presenta en 4.2.3, de una forma general, un CPLEX con varias capas de software adicionales para dar mejor tratamiento a este tipo de problemas, donde CPLEX en cualquiera de sus versiones hasta la fecha se queda corto. El uso del solver propuesto por los autores, requirió el modelamiento del GCSP bajo el marco RCSP, el cual se presenta en 4.2.3, utilizando el problema de enrutamiento de vehículos con ventanas de tiempo, carga y descarga presentado el manuscrito. En las comparaciones siguientes, este modelo aportado e implementado en este trabajo, será referido como **VRPSolver20**.

A causa de que Derigs and Schäfer (2014) para realizar el estudio comparativo resuelve el GCSP con el número de conductores como los fija Beasley et al. (1996), dicho trabajo no es suficiente para el análisis y experimentación realizados en esta investigación, sin embargo, como base de todo el estudio, en las Tablas 5.1 y 5.2, se presentan los resultados con el número de

conductores fijo y minimizando el número de conductores respectivamente.

Bloques de Trabajo	Beasley96			Derigs14				VRPSolver20			
	Número de empleados	Tiempo Cómputo (s)	FO	Número de empleados	Tiempo Cómputo (s)	FO	GAP	Número de empleados	Tiempo Cómputo (s)	FO	GAP
50	27	2	3139	27	0,60	3139	0,00	27	0,89	3139	0,00
100	44	8	4812	44	0,90	4812	0,00	44	0,94	4812	0,00
150	69	150	6275	69	1,00	6275	0,00	69	1,12	6275	0,00
200	93	61	6914	93	1,00	6914	0,00	93	1,12	6914	0,00
250	108	205	8406	108	2,00	8406	0,00	108	1,34	8406	0,00
300	130	260	9580	130	1,00	9580	0,00	130	1,61	9580	0,00
350	144	459	10991	144	2,00	10991	0,00	144	1,82	10991	0,00
400	159	17176	12341	159	8,00	12341	0,00	159	2,66	12341	0,00
450	182	17835	12785	182	8,00	12785	0,00	182	2,95	12785	0,00
500	204	1545	13302	204	7,00	13302	0,00	204	3,70	13302	0,00
Promedio		3770,10			3,15				1,82		
Sumatoria		37701			31,50				18,15		

Tabla 5.1: Comparativo metodologías con número de conductores Fijo K

BT	Beasley96			Derigs14			Modelo Completo						GC			VRPSolver20 - FO Costos			VRPSolver20-FO Min K					
	K	TC (s)	FO	K	TC (s)	FO	Heurística			Modelo (Inicial Heurística)			K	TC (s)	GAP	K	TC (s)	FO	GAP	K	TC (s)	GAP		
							K	TC (s)	FO	K	TC (s)	FO											K	TC (s)
50	27	2	3139	27	0,60	3139	0,00	0,00	0,00	27	2,72	3139	0,00	27	0,97	0,00	27	0,87	3139	0,00	27	0,89	0,00	
100	44	8	4812	44	0,90	4812	0,00	0,04	6981	0,02	42	3600,01	5458	-0,05	42	1,21	-0,05	42	0,91	5458	-0,05	42	1,06	-0,05
150	69	150	6275	69	1,00	6275	0,00	0,14	8342	0,03	67	3600,02	7213	-0,03	67	1,17	-0,03	67	0,97	7164	-0,03	67	0,99	-0,03
200	93	61	6914	93	1,00	6914	0,00	0,18	9686	-0,04	85	3600,02	9255	-0,09	85	4,31	-0,09	84	1,31	9679	-0,10	84	1,1	-0,10
250	108	205	8406	108	2,00	8406	0,00	0,42	14468	-0,01	105	3600,06	10193	-0,03	105	1,91	-0,03	105	1,51	9383	-0,03	105	13,08	-0,03
300	130	260	9580	130	1,00	9580	0,00	2,04	15272	0,02	130	3600,08	10304	0,00	130	1,82	0,00	130	1,62	9580	0,00	130	279,04	0,00
350	144	459	10991	144	2,00	10991	0,00	1,51	16208	-0,03	138	3600,04	14527	-0,04	134	8,20	-0,07	134	8,18	13389	-0,07	134	649,98	-0,07
400	159	17176	12341	159	8,00	12341	0,00	2,23	18385	-0,01	153	3600,05	17484	-0,04	151	113,88	-0,05	149	114,28	15118	-0,06	149	159,44	-0,06
450	182	17835	12785	182	8,00	12785	0,00	7,34	19174	0,01	184	3600,03	19174	0,01	177	5,94	-0,03	177	2,94	13717	-0,03	177	451,05	-0,03
500	204	1545	13302	204	7,00	13302	0,00	5,36	19945	-0,03	197	3600,02	19945	-0,03	183	127,77	-0,10	182	112,24	18134	-0,11	182	1283,91	-0,11
Promedio		3770,10			3,15			1,93		0,00		3240,30		-0,03		26,72	-0,04		24,48		-0,05		284,05	-0,05
Sumatoria		37701			31,50			19,28		-		32403,04		-		267,18	-		244,83		-		2840,54	-

Tabla 5.2: Comparativo metodologías minimizando el número de conductores **K**. Los GAPS son relativos Beasley96 y Derigs2014

5.2.1. Comparativo metodologías enfoque matemático

Las Tablas 5.1 y 5.2 presentan la información completa de los experimentos, sin embargo, en esta sección se busca discutir los más relevante que surge al comparar tiempos de cómputo, calidad de las soluciones de las metodologías de enfoque matemático, y el comportamiento equilibrado de la propuesta de solución para esta tesis que es de carácter aplicado.

Inicialmente, se tiene en la Figura 5.2 el comparativo de las metodologías en términos de tiempo. En este análisis no es posible consignar todas las implementaciones, porque los estudios previos a pesar de trabajar el GCSP, difieren en el tratamiento del número de turnos como se ha mencionado previamente. Lo que sí es posible, es que utilizando el **VRPSolver20** con un número fijo de conductores, se puede conocer el rendimiento de esta metodología frente a la literatura especializada y servir como referente para el **GC** y el impacto de una función objetivo orientada a minimización de conductores en el mismo **VRPSolver20**. En 5.2 se observa el gran rendimiento que tiene el modelamiento propuesto en este trabajo del GCSP en este solver especializado (sección 4.2.3), al comparar el tiempo total de cómputo de las implementaciones para el total de la librería <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/cspinfo.html>. A pesar de la normalización de los tiempos de cómputo de Beasley et al. (1996) como se presenta en Derigs and Schäfer (2014) utilizando los estudios publicados en Dongarra, Jack J (2009), existe una importante brecha. Derigs and Schäfer (2014) por otro lado, presenta la solución más eficiente y reciente de la literatura especializada, pero el tiempo de cómputo es mejorado (disminuído) por el **VRPSolver20**. Finalmente en 5.2 no se presenta el experimento realizado con el **Modelo Completo** con un número de conductores fijo, porque muchas de las instancias presentan un GAP que descalifican el uso del CPLEX (incluso en sus versiones recientes) sin mecanismos especializados para abordar este

problema.

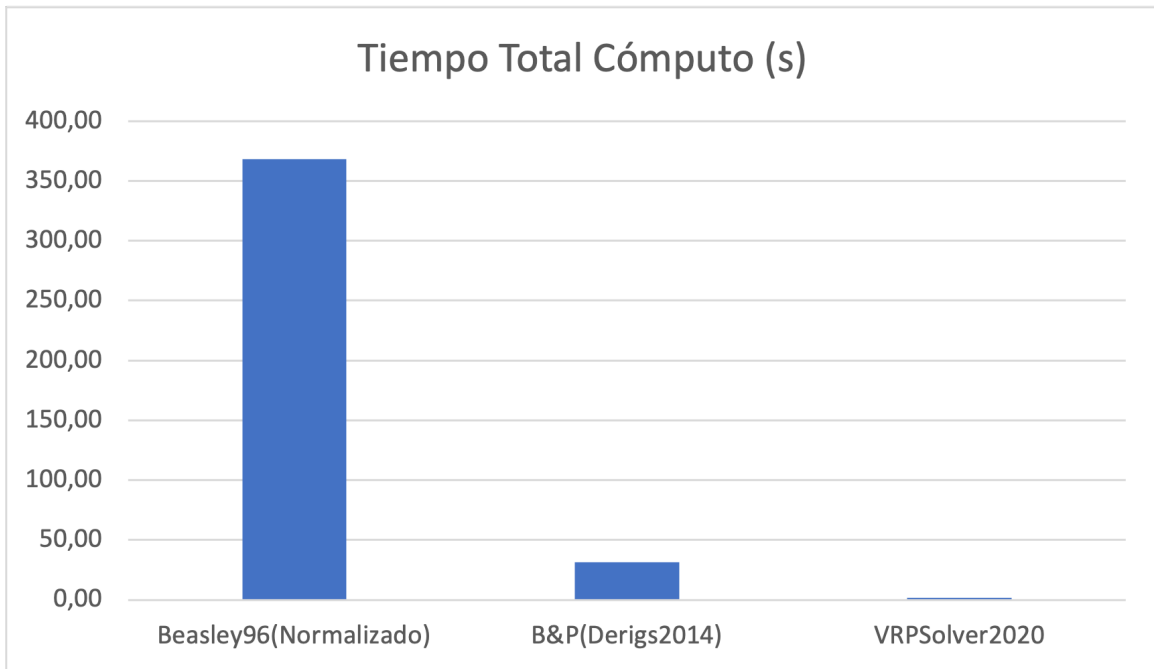


Figura 5.2: Tiempo de cómputo librería CSP genérico completa, con número fijo de conductores. Fuente: Elaboración propia.

Ahora bien, una vez analizado desde la perspectiva de los tiempos de cómputo, las metodologías se comparan en la Figura 5.3 bajo el interés principal de este trabajo, la minimización de conductores.

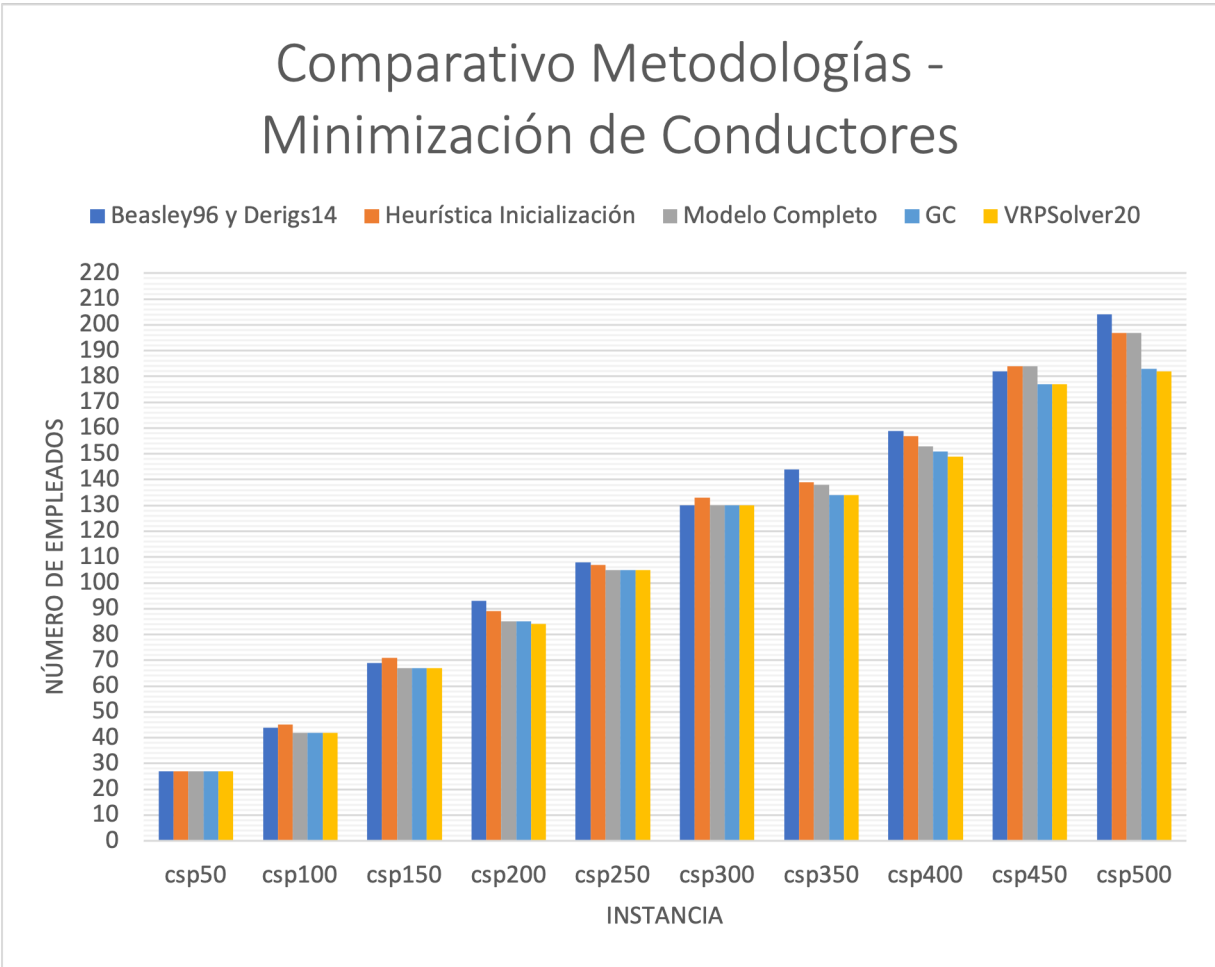


Figura 5.3: Comparativo minimización de número de conductores. Fuente: Elaboración propia.

Aunque los trabajos de **Beasley96** y **Derigs14** manejan un número fijo de conductores en el componente principal de la metodología, ambos están en función de una exploración del número de turnos, bien sea recibiendo el número como un input o realizando una especie de búsqueda binaria hacia arriba y hacia abajo. La Figura 5.3 compara entonces todas las metodologías frente a la minimización de itinerarios, mostrando varias situaciones interesantes que se listan a continuación:

- El **VRPSolver20** y el **GC** propuestos en este trabajo logran el uso mínimo de los operadores en todas las instancias de prueba, y aportando a la literatura los nuevos BKS para el GCSP.

En la sección A.1 están especificados los BKS obtenidos en este estudio.

- La **Heurística de Inicialización** para el **Modelo Completo** en muchos de los casos empata o incluso mejora a **Beasley96** y **Derigs14** en cuanto al número de conductores empleados y con tiempos de cómputo muy bajos como se aprecia en la Tabla 5.2, tornándose interesante como una metodología muy rápida, y que gracias a este estudio, en el cual se han encontrado los óptimos o BKS para las instancias, con certeza podemos decir que son de muy buena calidad, arrojando un balance muy interesante en términos prácticos entre facilidad de implementación, tiempo de cómputo y calidad.
- El **Modelo Completo** tiene unos resultados interesantes en cuanto a la minimización de conductores, porque en todos los casos supera a la literatura especializada, sin embargo, presentando tiempos de cómputo altos (límite de tiempos establecido para los experimentos en la mayoría de los casos), la mejora que logra sobre el punto inicial recibido de la **Heurística de Inicialización**, deja una proporción costo/beneficio que no es interesante para soluciones prácticas.
- En la medida en que las instancias aumentan su tamaño, se observa una diferencia relevante entre las metodologías de la literatura especializada y el aporte que se realiza en esta investigación.

Luego de observar los buenos resultados de las metodologías desarrolladas en este trabajo, es necesario revisar cuál de todas presenta el mejor rendimiento en términos del tiempo computacional. En la Figura 5.4 se presenta para cada una de las instancias, el tiempo de cómputo de los métodos **GC** y **VRPSolver20**. De este último, se presentan dos versiones, **VRPSolver20 - FO Costos** que corresponde a la propuesta original que hereda la penalización reactiva propuesta en el **Modelo Completo**, y un experimento adicional que se realizó, minimizando el número de

aristas salientes del depósito, denotado como **VRPSolver20 - Mín Conductores**. Este último buscó observar el comportamiento de una función objetivo compacta que ayudara en la exploración del espacio de soluciones, y que además, recreara condiciones para una comparación directa con el **GC**. Bajo este contexto, que es el que más se asemeja a la práctica, donde no se cuenta con unos costos parametrizados de forma explícita, el **GC** presenta el mejor rendimiento, superando al **VRPSolver20 - Mín Conductores**, el cual se torna inestable y tiene dificultades de convergencia cuando se incrementa el número de servicios que se requiere programar como se observa en la Figura 5.4.

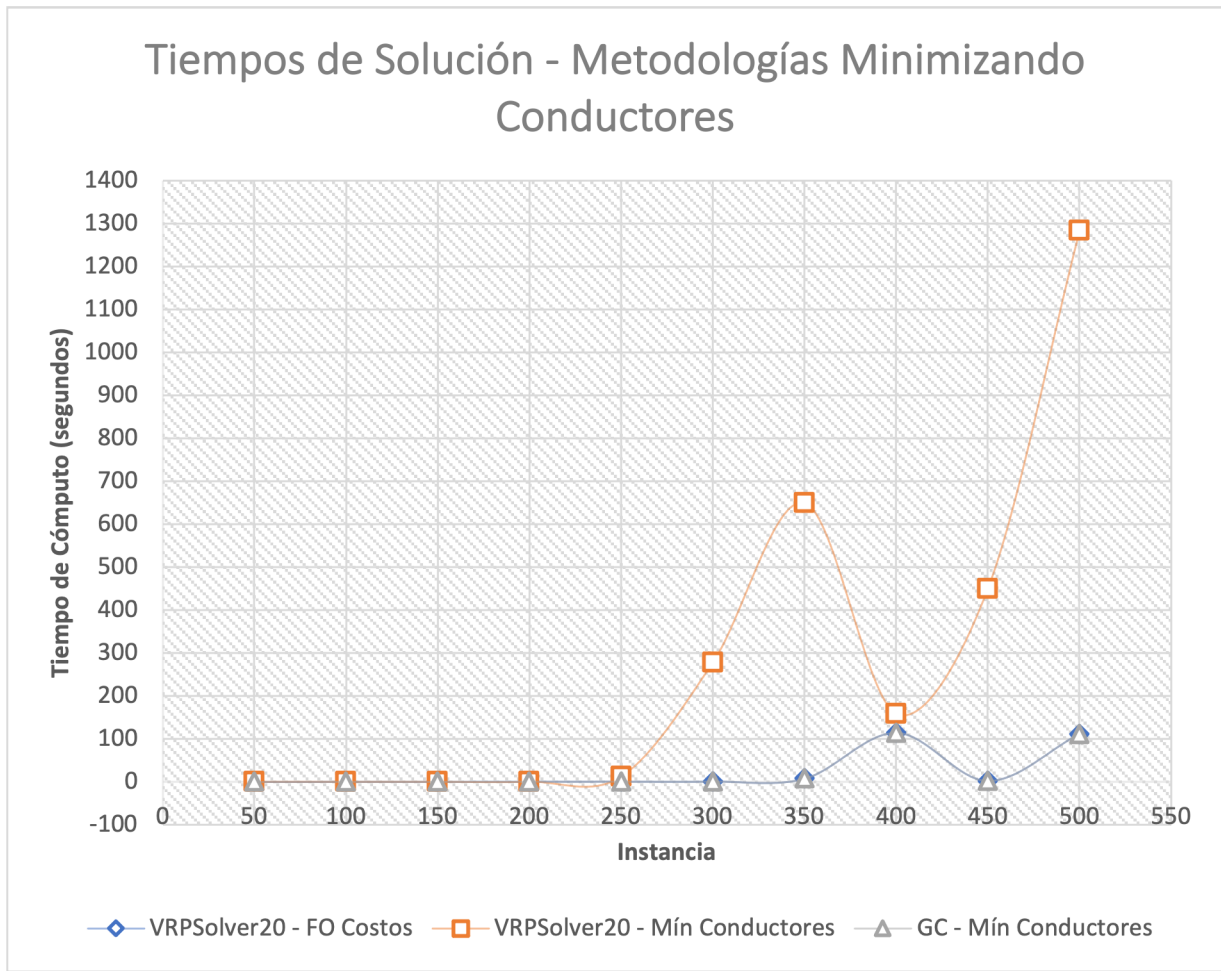


Figura 5.4: Comparativo metodologías que alcanzan BKS para número de conductores. Fuente: Elaboración propia.

La Figura 5.4, permite apreciar las particularidades del espacio de soluciones del problema abordado en este trabajo, metodologías muy similares, que involucran generación de columnas y aplicación de cortes, al modificar la función objetivo, presentan una divergencia notoria en sus tiempos de cómputo. Especialmente, la instancia *csp400*, se ve favorecida cuando se utiliza una función objetivo que minimiza el número de conductores únicamente; mientras que las funciones objetivo basadas en costos interbloque, y costos reactivos por cada turno generado, presentan dificultades guiando la metodología para este caso, saliéndose de la tendencia del repositorio como se muestra en el gráfico comparativo citado. Esto, se debe a que en la mayoría de los casos, para un número mínimo de conductores K , existen muchas posibles soluciones, todas con costos interbloque diferentes, sin embargo, el caso *csp400*, al tener un espacio de soluciones más restringido, ofrece menos configuraciones factibles con un K mínimo, favoreciendo el uso de una función objetivo que minimiza únicamente el número de conductores. En contraste, la instancia *csp350*, la cual cuenta con un número elevado de soluciones para el K mínimo, genera problemas cuando la función objetivo se centra en la minimización del número de turnos K . En términos generales, se aprecia que involucrar los costos de abrir un itinerario, y los costos de cambio de contexto entre bloques, brindan una mayor sensibilidad para que las metodologías exploren de una forma más eficiente el espacio de soluciones.

5.2.2. Marco general y caso real

Habiendo completado el estudio computacional, el **GC** es la solución más estable y con mejor rendimiento en las instancias de la literatura especializada. Como experimento final, en el cierre de la investigación fue posible asignar la demanda de servicios establecidos por el ente gestor del

sistema de transporte masivo de la ciudad, correspondiente al 10 de marzo de 2021, disminuyendo en un turno la asignación de servicios a los operadores. La metodología de dos fases desde las tablas de servicio hasta la constitución de los turnos como se ilustra en la Figura 5.5 fue realizada.

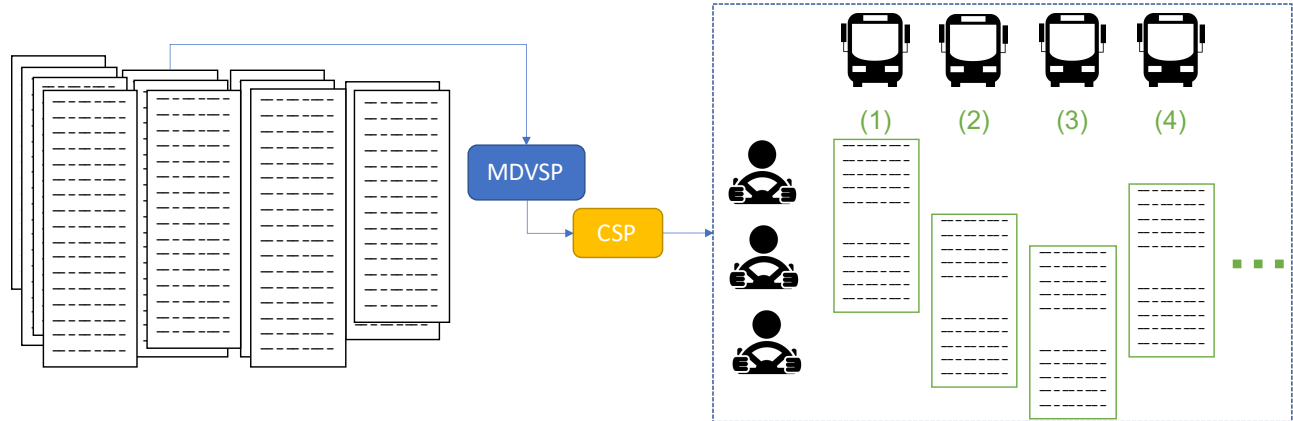


Figura 5.5: Metodología Integrada - Sistema de Transporte Masivo. Fuente: Elaboración propia.

En el frente del MDVSP, el centro de operaciones utiliza la implementación del algoritmo [Moreno et al. \(2019\)](#), el cual optimiza la utilización de la flota disponible, disminuyendo el tiempo de ocio y dejando itinerarios muy compactos que liberan parte de la flota para realizar la rotación de los vehículos como se muestra en la Figura 5.6 presentada a continuación.

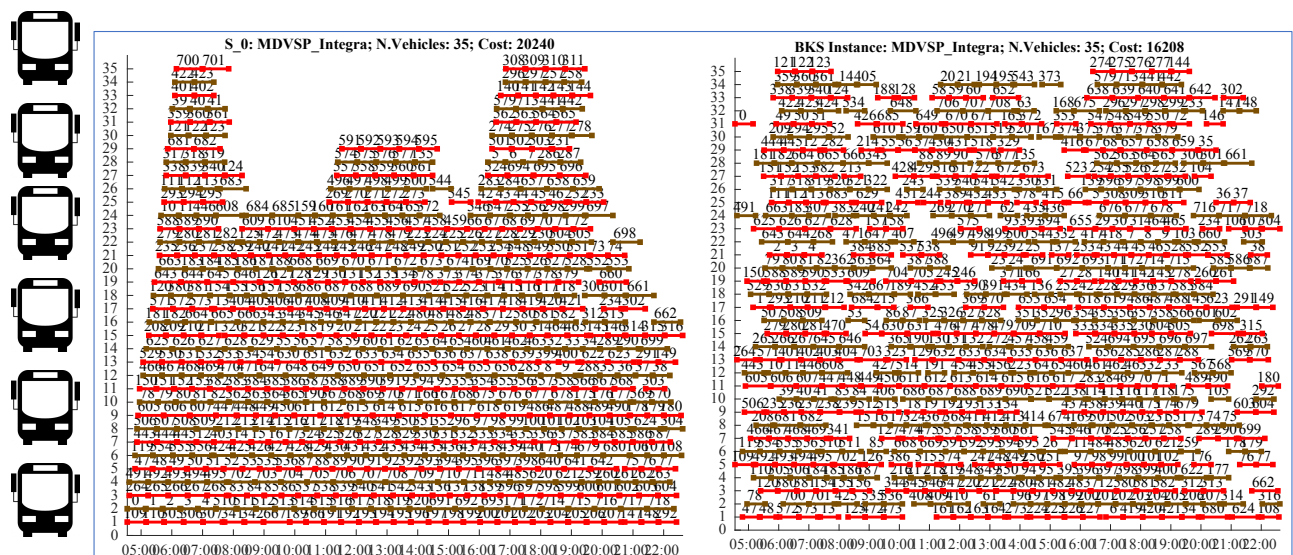


Figura 5.6: Efecto de la primera fase de la metodología. Tomado de: [Marín-Moreno \(2019\)](#).

En la segunda fase, se utiliza el algoritmo de **GC** y el balance fue el siguiente:

- **Número de Turnos:** 99
- **Turnos de 3 Bloques:** 73
- **Turnos de 2 Bloques:** 25
- **Turnos de 1 Bloques:** 1
- **Tiempo de 1 Cómputo:** 102 segundos
- **Número de Servicios:** 3564 (Tamaño del caso)

El resultado obtenido contrasta con el tiempo de cómputo manual de los expertos, quienes requieren de media jornada (4 horas) para generar una programación de conductores. Se destaca el gran conocimiento del centro de control del problema, quienes generan un turno adicional frente al **GC**. Como se observa, el número de servicios de la operación no es trivial, sin embargo, el tiempo de cómputo es corto, y no presenta un crecimiento exponencial al superar con creces el caso más grande de la literatura especializada, debido a que las restricciones técnicas y de bienestar laboral hacen que el espacio de soluciones sea manejable.

Capítulo 6

Conclusiones, trabajos futuros y producción

6.1. Conclusiones

El cierre de esta tesis doctoral, planeada como un proyecto de investigación aplicada, genera frutos en ambos frentes; aportando a la literatura especializada modelos y metodologías que presentaron un rendimiento adecuado tanto en instancias clásicas del problema, así como en la operación real de un sistema de transporte. Este último, lo suficientemente grande como para poner a prueba algoritmos de optimización especializados.

Una primer hallazgo que arroja el estudio realizado, y es importante que sea tenido en cuenta para investigadores futuros que estén interesados en el tema, es la dificultad para conseguir casos de estudio generales, que además presenten variación y dimensión suficientes para enriquecer los experimentos. De la misma forma, las pocas instancias disponibles en los repositorios presentan tres inconvenientes: (i) como sucede en el trabajo [Huisman et al. \(2005\)](#), donde se resuelve el MDVSP y el CSP de manera conjunta, las instancias requieren de unas restricciones muy específicas del sistema de transporte en el cual se basó el estudio; (ii) algunas instancias de

estudios relevantes para el componente CSP, son información reservada por las empresas del sector transporte que patrocinan estos desarrollos, entre otras razones, porque las soluciones que arrojan estas metodologías son un factor diferenciador en la planeación operativa; presentando en las publicaciones los buenos resultados alcanzados, pero sin los detalles de los casos de estudio. En tercer lugar (iii), la librería identificada con instancias lo suficientemente generales, no contaba con experimentos computacionales completos, al no considerar la minimización del número de conductores en ninguna de las publicaciones que utilizan este repositorio. Las instancias de la librería, modelan el problema con un comportamiento multiobjetivo, difícilmente aplicable a la práctica, dado que plantean que el costo general de la programación, se disminuye cuando se emplean más conductores, porque se incluyen menos transiciones o cambios de contexto entre los bloques de trabajo que constituyen los turnos. Este enfoque resulta poco práctico, porque implica fijar el número de conductores para minimizar las soluciones, además de distar mucho de la realidad, donde es clave minimizar la ocupación del recurso humano con el que se cuenta, por aspectos como: bienestar laboral, disponibilidad de operadores para contingencias y costos operativos.

Estas particularidades identificadas en la exploración de repositorios especializados sin embargo, trajo consigo la generación en este trabajo de dos productos adicionales relevantes: un modelo matemático que formula de manera completa el GCSP, ausente en la literatura especializada, y el modelamiento de una solución bajo el concepto de un solver especializado, en ambos casos, abordando de manera exitosa la minimización de conductores, en el marco de dos fases de este proyecto.

En la fase final de esta tesis, se identifica una tendencia en la literatura especializada,

consistente en la integración de una interfaz de modelamiento en un lenguaje de programación interpretado y de alto nivel para la carga de datos y construcción del modelo matemático, que entregan a una implementación especializada en un lenguaje compilado de alto rendimiento, las restricciones, función objetivo y parámetros. En [Pessoa et al. \(2020\)](#) específicamente, se utiliza el lenguaje de programación Julia como interfaz para modelar el problema bajo el marco RCSCP descrito en la [4.2.3](#), y entregado a una implementación realizada en C++ que emplea CPLEX, Boost y desarrollos específicos para este tipo de problemas. Los experimentos de este trabajo ratifican la efectividad de esta forma de solución y modelamiento.

El **GC** modela situaciones realistas de la operación, sin perder generalidad. Sus restricciones en cuanto a transiciones y constitución de los turnos, bajo la mayoría de las regulaciones laborales son aplicables. La propuesta de este trabajo muestra ser un aporte a la literatura, mientras que genera los resultados de mayor impacto de la investigación al resolver el problema real.

No todas las investigaciones logran reportar óptimos o BKS al ser desarrolladas, y más aún cuando trabajan problemas clásicos que han sido formulados hace varias décadas y hacen parte de la espina dorsal de la investigación de operaciones, como es el caso del GCSP. Este trabajo logra con dos de sus propuestas encontrar nuevas soluciones para una librería publicada en 1996. En este trabajo se anexan en la sección [A.1](#) los óptimos generales en cuanto a costos interbloqueo y el número mínimo de conductores necesarios para atender los servicios.

Esta investigación utiliza una metodología especializada desarrollada en la unidad de investigación de la empresa para la fase de programación de vehículos, y centra sus esfuerzos en desarrollar un marco de trabajo especializado adecuado para la programación de conductores. Este último, con más restricciones técnicas, presenta un espacio de soluciones difícil de explorar,

el cual requirió de la experimentación con muchos enfoques para encontrar la manera adecuada de abordarlo. La conclusión más relevante del estudio comparativo, a pesar del alto rendimiento del modelo implementado con el **VRPSolver** en las instancias de GCSP, es que el **GC** propuesto en este trabajo presenta un rendimiento similar al **VRPSolver**, pero logrando además incluir todas las restricciones técnicas solicitadas por el centro de operaciones de la organización. Sumado a eso, el **GC** no requirió de costos reactivos para cada instancia que penalizaran la ocupación de nuevos conductores, su estructura naturalmente minimiza el número de operadores, y el costo interbloque entre los bloques de trabajo. En contraste, el **VRPSolver** se ve afectado si no se explora el espacio de soluciones con costos calculados, dado que el problema puede presentar múltiples soluciones con un número mínimo de conductores, pero con turnos que presentan cambios de contexto costosos.

Los modelos inmersos en las metodologías híbridas presentadas en esta tesis, ofrecen formas alternativas a la literatura especializada para trabajar el problema. El **VRPSolver** sin embargo, para contextos empresariales, requiere el pago no solamente de la licencia del solver, si no también, del módulo que genera los callbacks y columnas especializadas, haciéndolo menos viable en la práctica que la **GC** desarrollada.

Con todo lo anterior, los aspectos de planeación operativa concernientes a la programación de vehículos y programación de conductores, al cierre de esta investigación, han sido mejorados de forma sobresaliente en la organización, y de manera relevante en el estado del arte.

6.2. Trabajos futuros

Al explorar los diferentes frentes del problema tratado, se identifican experimentos que serían de gran utilidad para el estado del arte asociado al marco de planeación operativa trabajado. A continuación se enumeran los más relevantes:

1. Así como el **GC** fue adaptado a las instancias de prueba con el fin de medir su rendimiento, los buenos resultados del **VRPSolver20** como sustento del estudio computacional para la librería especializada de CSP genérico, se sugiere incorporar las restricciones prácticas al modelamiento del problema en el **VRPSolver20** para que resuelva el caso real. Resultados de gran interés podrían obtenerse tanto para el sector productivo como para el estado del arte.
2. Las variantes al problema GCSP propuestas en [Derigs and Schäfer \(2014\)](#), una minimizando además los tiempos de ocio entre bloques de trabajo, y otra analizando un horizonte de tiempo de 48 horas, sería interesante modelarlas y resolverlas con las metodologías **VRPSolver20** y **GC** propuestas en este trabajo, dado que este trabajo ofrece una base para los comparativos de rendimiento, y estos aspectos son interesantes para aplicaciones reales de transporte.
3. Este trabajo hace parte de un proyecto mayor que busca realizar el totalidad de la planeación operativa del sistema de transporte masivo del Área Metropolitana Centro Occidente con algoritmos de optimización especializados, para realizar la toma de decisiones con criterios enriquecidos y altos niveles de eficiencia en el uso de la flota, el personal y la cobertura de servicios. El trabajo presentado consistió en la integración de la metodología desarrollada en [Moreno et al. \(2019\)](#) que resuelve el problema del MDVSP, con el desarrollo de una

metodología especializada para la gestión del personal del sistema BRT estudiado. El paso siguiente del marco general, consiste en agregar el horizonte de tiempo al personal que realiza los turnos construídos, es decir, el problema de rostering como se establece en [Ceder \(2007\)](#) y se ilustra como una tercera etapa en la Figura 4.2. En la empresa se han dado los primeros pasos que pueden servir como punto de partida al trabajo futuro recomendado en este numeral, y se puede encontrar en [Parra \(2019\)](#).

Los trabajos enumerados podrían llevarse a cabo en trabajos de maestría y doctorado, aportando a la comunidad académica y empresarial, interesadas en los problemas de asignación de servicios a flota y tripulaciones de diferente índole.

6.3. Producción

En la presente sección, se listan los productos académicos, fruto del desarrollo y difusión de esta investigación:

1. **Publicación en Revista Especializada Q1:** Marín-Moreno, C.A., Escobar-Falcón, L.M., Bolaños, R., Subramanian, A., Zuluaga, A., Granada, M. (2019). A hybrid algorithm for the multi-depot vehicle scheduling problem arising in public transportation. *International Journal of Industrial Engineering Computations*, 10(3), 361-374.
2. **Publicación en Revista Especializada Q1:** Bolaños, R., Escobar, J.W., Granada, M. (2018). A metaheuristic algorithm for the multi-depot vehicle routing problem with heterogeneous fleet. *International Journal of Industrial Engineering Computations*, 9(4), 461-478.
3. **Publicación en Revista Especializada Q1:** Bolanos, R., Toro, E.M., Granada, M. (2016). A

- population-based algorithm for the multi travelling salesman problem. *International Journal of Industrial Engineering Computations*, 7(2), 245-256.
4. **Publicación en Revista Especializada Q3:** Marín-Moreno, C., Bolaños, R., Escobar-Falcón, L.M., Escobar-Velásquez, J.W. (2020). Constructive matheuristic algorithms for solving the multi-depot vehicle scheduling problem for public transportation. *Revista Espacios*, ISSN: 0798-1015, Vol. 41 (27) 2020, 174-191.
 5. **Publicación en Revista Especializada C(Pubindex):** Escobar-Falcón, L. M., Cárdenas, K., Marín-Moreno, C. A., Bolaños, R. I., Martínez, D. A. (2020). Elasticidad de la demanda en el subsistema de transporte masivo del Área Metropolitana Centro Occidente en función del incremento de la tarifa. *Entre Ciencia e Ingeniería*, 14(27), 63-72.
 6. **Capítulo de Libro:** Bolaños, R., Escobar-Falcón, L.M., Cárdenas, K., Granada, M. , Marín-Moreno, C.A.. Un algoritmo eficiente para la programación de turnos de trabajo en la operación de un sistema de transporte masivo. *AVANCES Y PERSPECTIVAS DE LA INGENIERÍA 4.0*. En: Colombia ISBN: 978-958-8859-60-6 ed: Universidad Libre , v. , p.380 - 403, 2020.
 7. **Capítulo de Libro:** Bolaños, R., Escobar-Zulua, A.H., Granada, M., Marín-Moreno, C.A., Escobar-Falcón, L.M., Subramanian, A..Algoritmo Híbrido para Solucionar el Problema de Programación de Vehículos de Transporte Público con Múltiples Depósitos. *Proceeding of the XIX Latin-Iberoamerican Conference on Operations Research, CLAIO 2018*. En: Perú ISBN: 978-612-48146-1-7 ed: Sociedad Peruana de Investigación Operativa (SOPIOS) , v. , p.85 - 92, 2019.
 8. **Publicación en Memorias de Congreso:** Giraldo, L., Marín, C., Bolaños, R. (2016). Design

of an architectural model suitable for Metropolitan West Central Area in Colombia. En *23rd ITS World Congress, Melbourne, Australia.*

Anexo A

A.1. Óptimos *GCSP* (Beasley, 96)

A continuación se presentan las soluciones óptimas, minimizando tanto número de conductores, como costo interbloque para la librería presentada por [Beasley et al. \(1996\)](#). En la literatura especializada, no se ha resuelto este problema minimizando ambos aspectos, constituyendo así, una de las principales contribuciones de este trabajo al estado del arte. En el presente apéndice, se explicitan las soluciones (Figuras [A.1](#) - [A.10](#)), las cuales podrán ser de utilidad para la realización de futuros experimentos, y estudios comparativos con nuevas propuestas metodológicas.

Las mencionadas Figuras [A.1](#) - [A.10](#), presentan el subgrafo solución de las instancias [Beasley et al. \(1996\)](#). Los problemas de programación de conductores CSP (Por sus siglas en inglés: *Crew Scheduling Problem*), son grafos incompletos, a causa de la indexación en el tiempo de los nodos, criterio principal para permitir las conexiones entre estos. Se muestra entonces en ellas, la solución de cada uno de los casos de la librería, las cuales consisten en un subgrafo de cobertura, resultado de la selección del número mínimo de caminos de mínimo costo que cubren todos los nodos, correspondientes a servicios de obligatorio cumplimiento por el operador, y por tanto, de los

tripulantes que están a cargo de esta labor.

Puntualmente, los subgrafos sucesivos, representan las soluciones de la siguiente manera:

- El centro de control (centro de operaciones), desde donde los conductores iniciarían sus itinerarios, está denotado por uno de los nodos cuadrados etiquetados con 0.

- Los arcos salientes en color verde del centro de control 0, representan el inicio del itinerario.

A cada uno de estos arcos se asocia: (i) un costo que corresponde a la ocupación de un nuevo operador para el turno, denotado por c , y (ii) el tiempo de servicio del nodo al cual se encuentra conectado, denotado por t . Es importante recordar, que estos arcos no hacen parte del input que ofrece la librería [Beasley et al. \(1996\)](#), y corresponden a la penalización reactiva propuesta en este trabajo, detallada en la sección [4.2.2](#).

- De forma análoga a los arcos descritos en el ítem anterior, los arcos interbloque son representados con color azul, y tiene asociados dos valores: (i) el costo c correspondiente al cambio de contexto entre bloques de trabajo del itinerario establecidos en la librería, y (ii) el tiempo t que denota la transición entre los nodos, junto con el tiempo de servicio del siguiente nodo, es decir, el tiempo transcurrido entre la finalización del nodo i y el comienzo del nodo j , más el tiempo de servicio del nodo j . Lo anterior, con el fin de mostrar en las soluciones el tiempo transcurrido, o carga de trabajo acumulada en cada uno de los bloques de trabajo que representan los nodos.

- Cada camino, desde el nodo 0, hasta el nodo terminal, constituye un turno o itinerario de un operador.

- El nodo inconexo de forma cuadrada, denotado por $N1$, simplemente representa el retorno

opcional al centro de operaciones por parte de los conductores al terminar su jornada, pero no constituye tiempo del turno, ni implicaría costos para la empresa operadora. Ha sido incluido en los subgrafos solución presentados en este apéndice, para guardar correspondencia con la notación presentada en [Beasley et al. \(1996\)](#), donde este retorno se representa con el nodo $N + 1$, pero sin aportar al tiempo o costo total del turno.

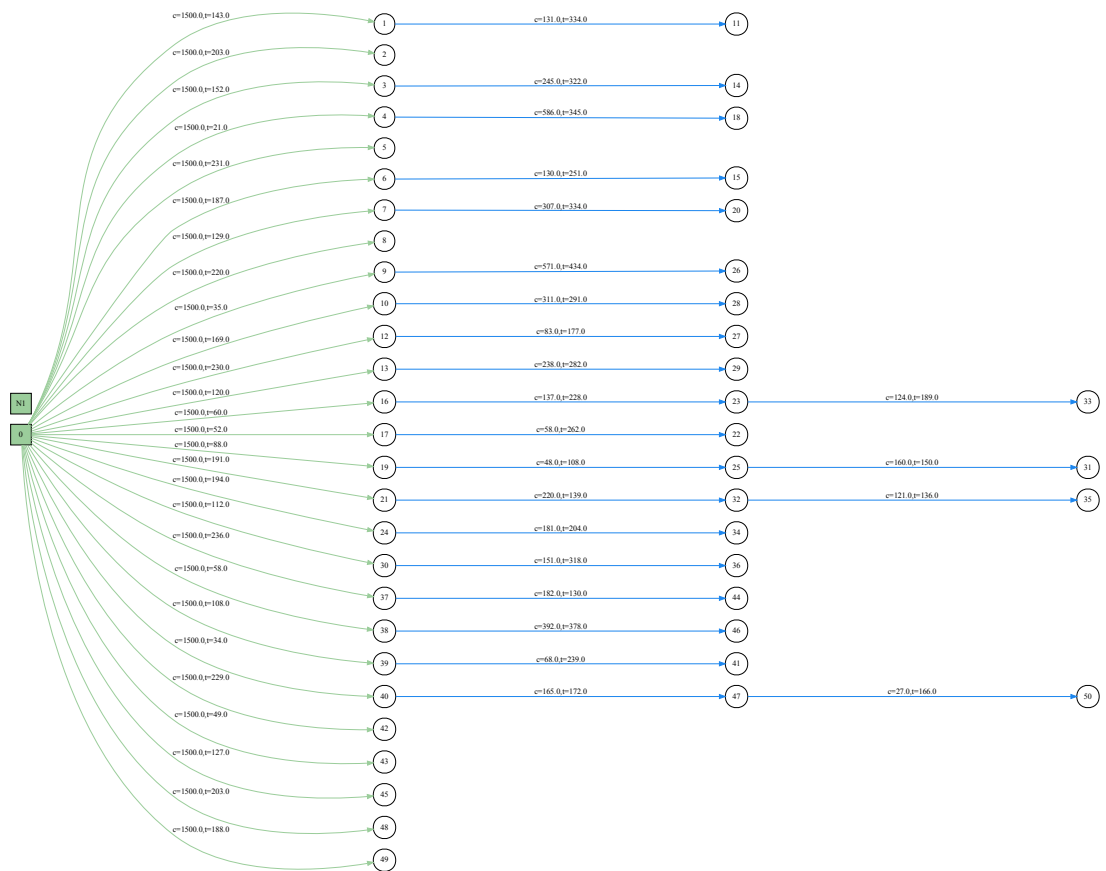
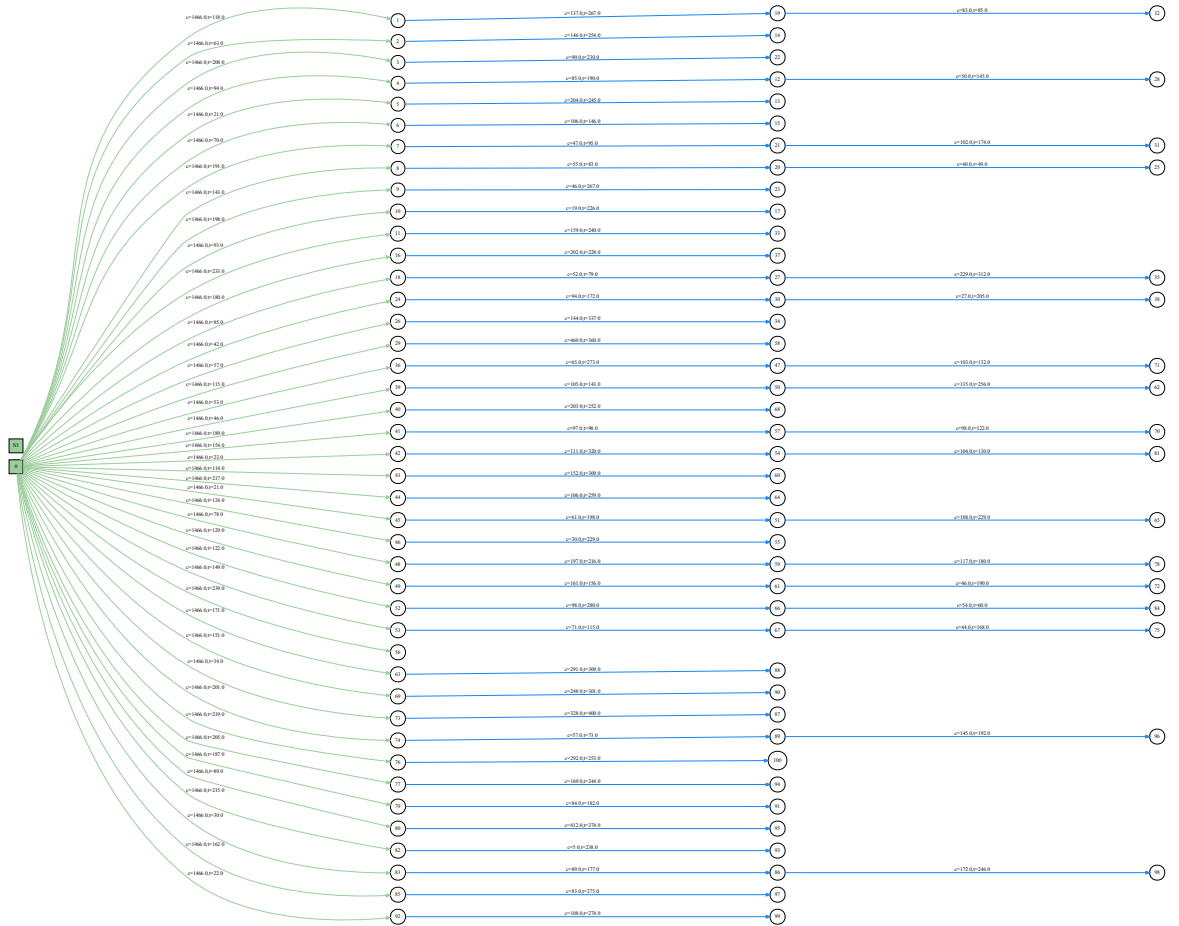


Figura A.1: BKS itinerarios *csp50*



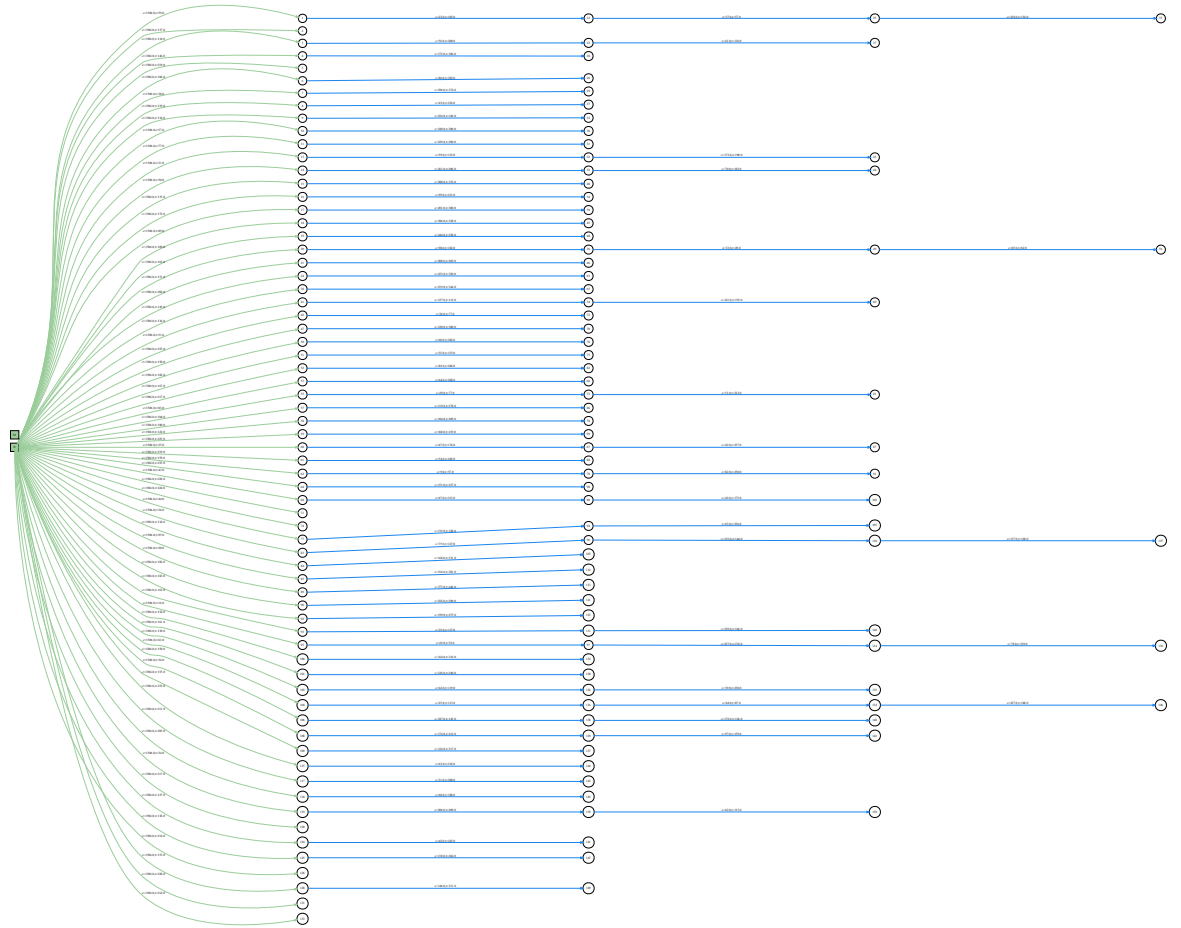


Figura A.3: BKS itinerarios *csp150*

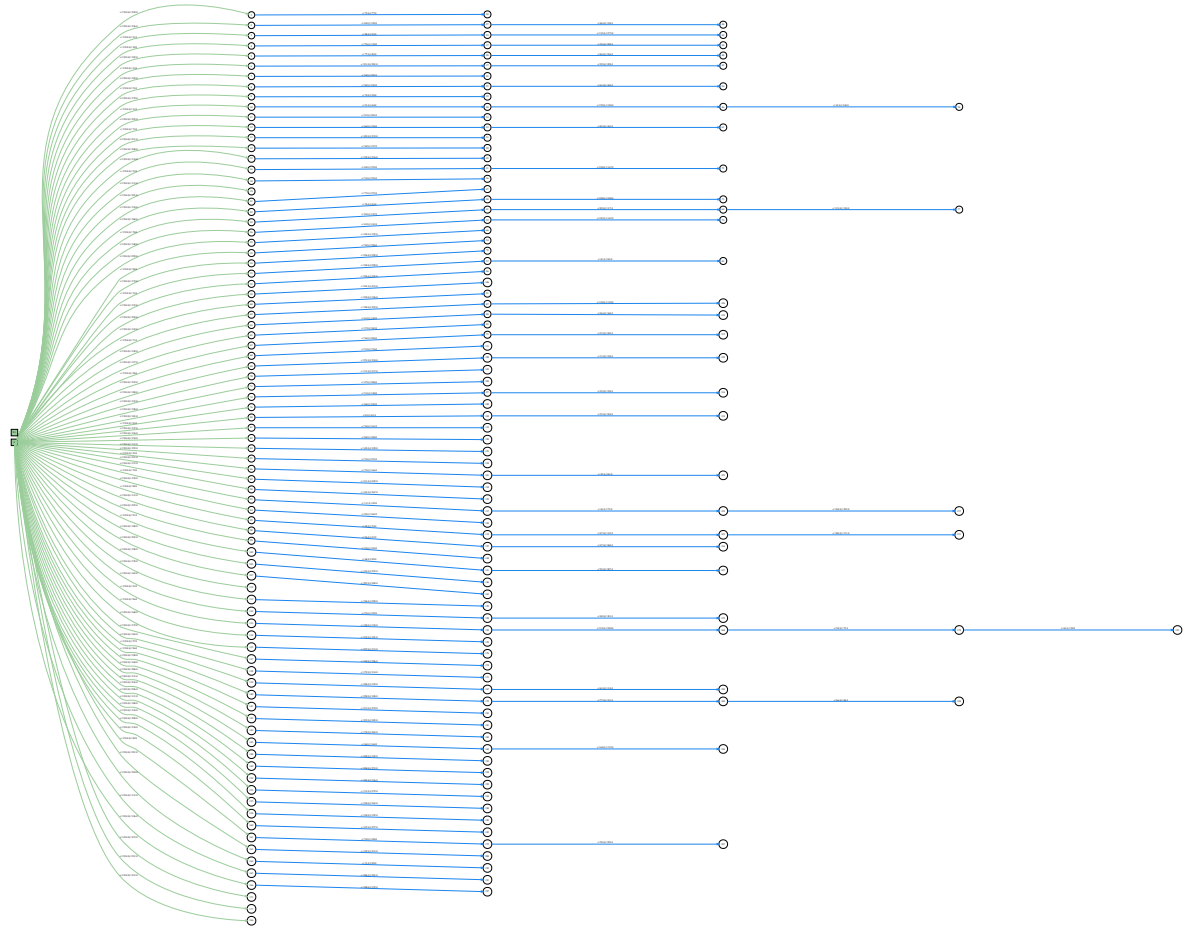


Figura A.4: BKS itinerarios *csp200*

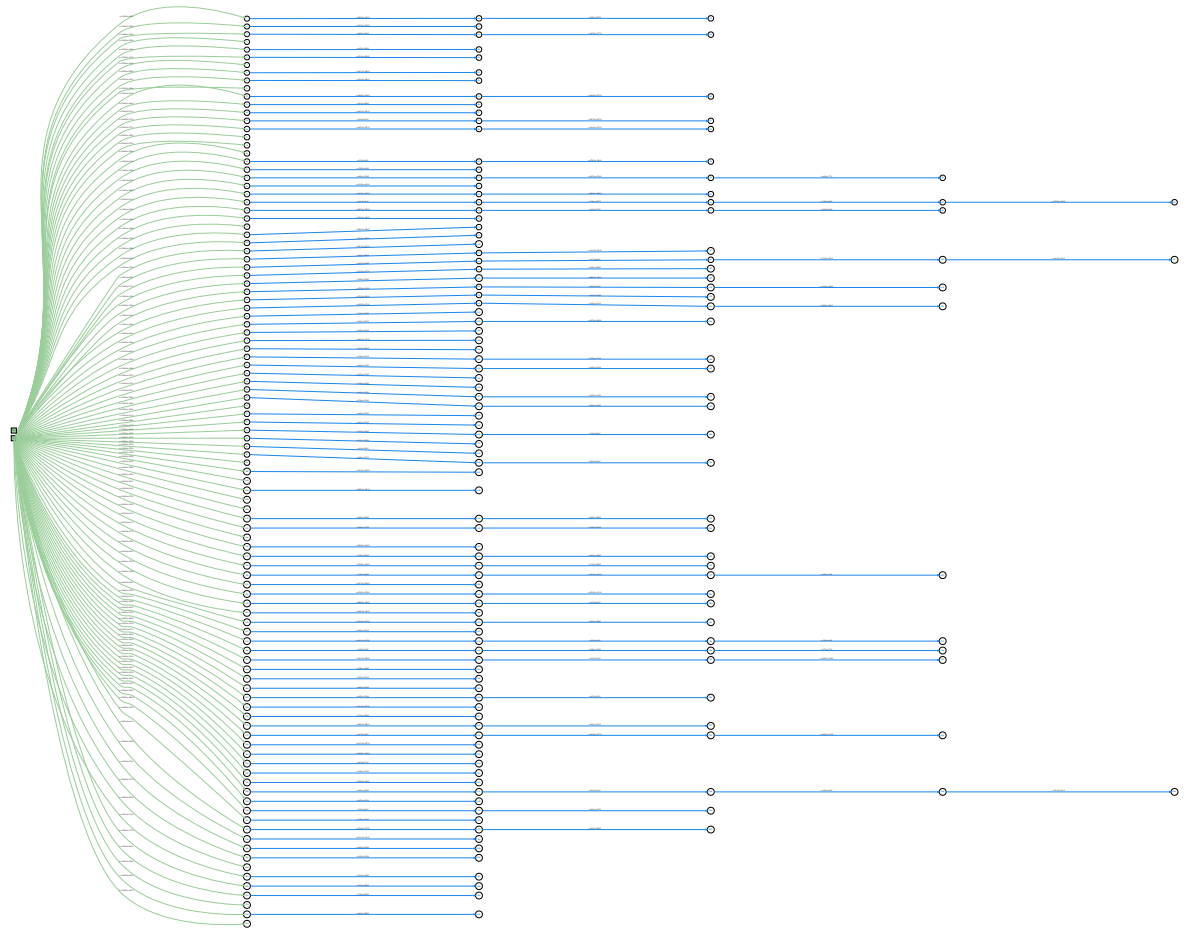


Figura A.5: BKS itinerarios *csp250*

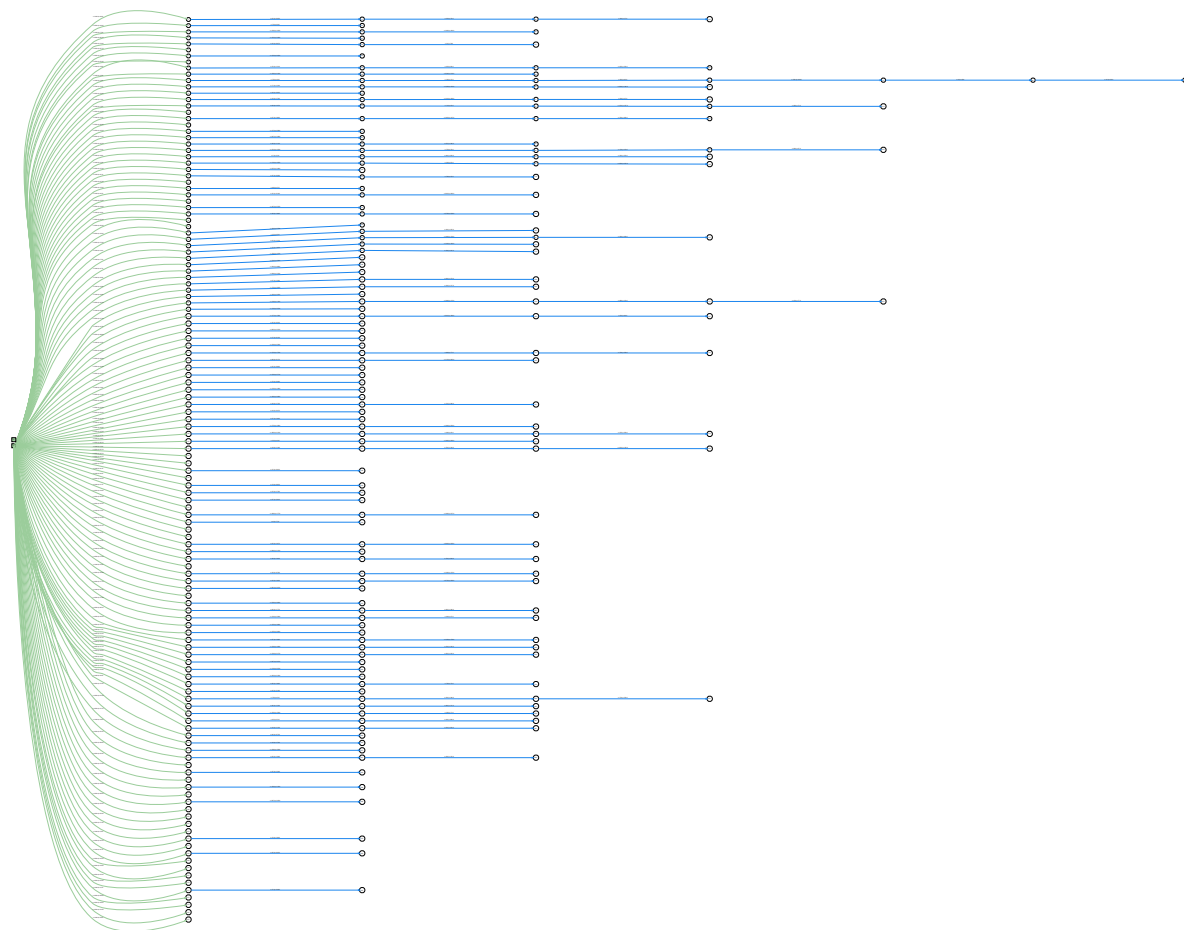


Figura A.6: BKS itinerarios *csp300*

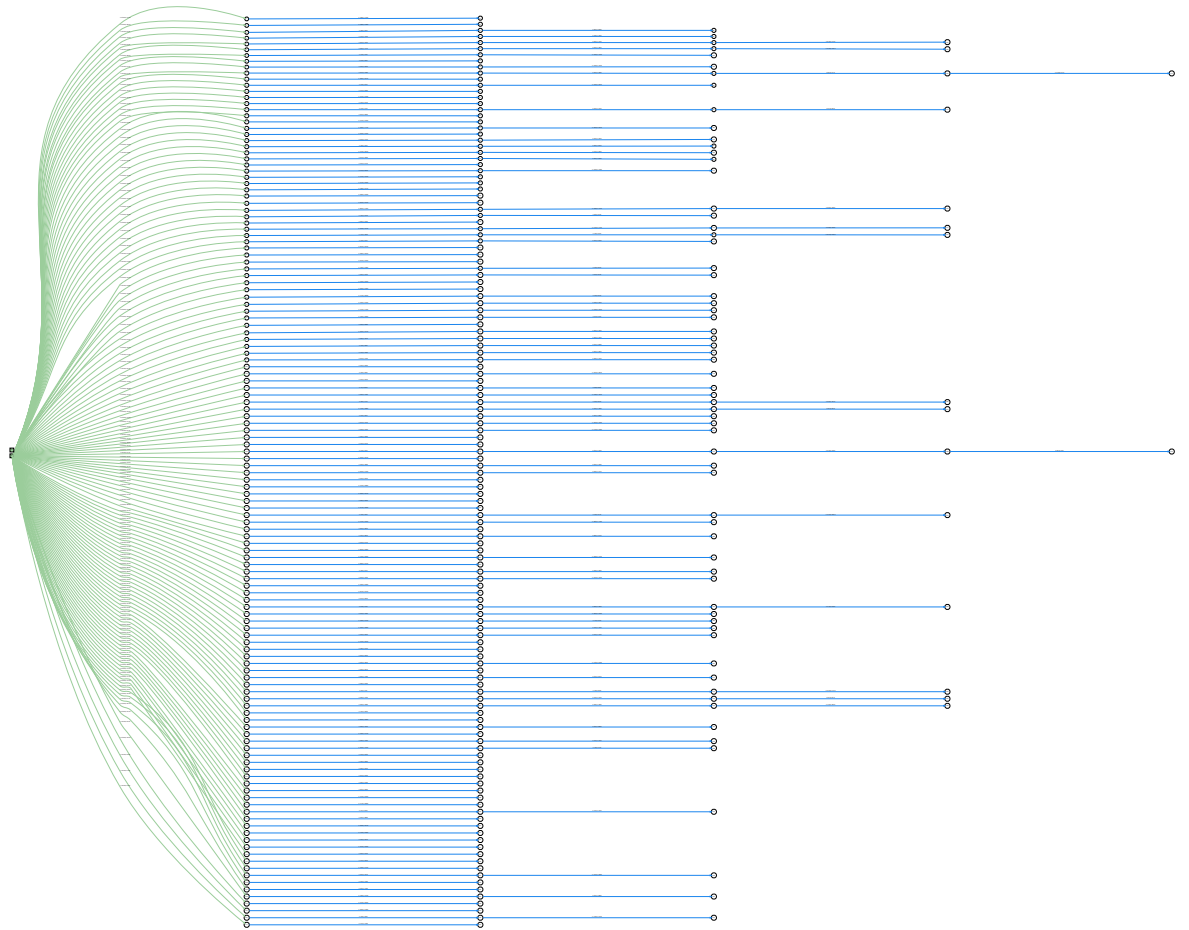


Figura A.7: BKS itinerarios *csp350*

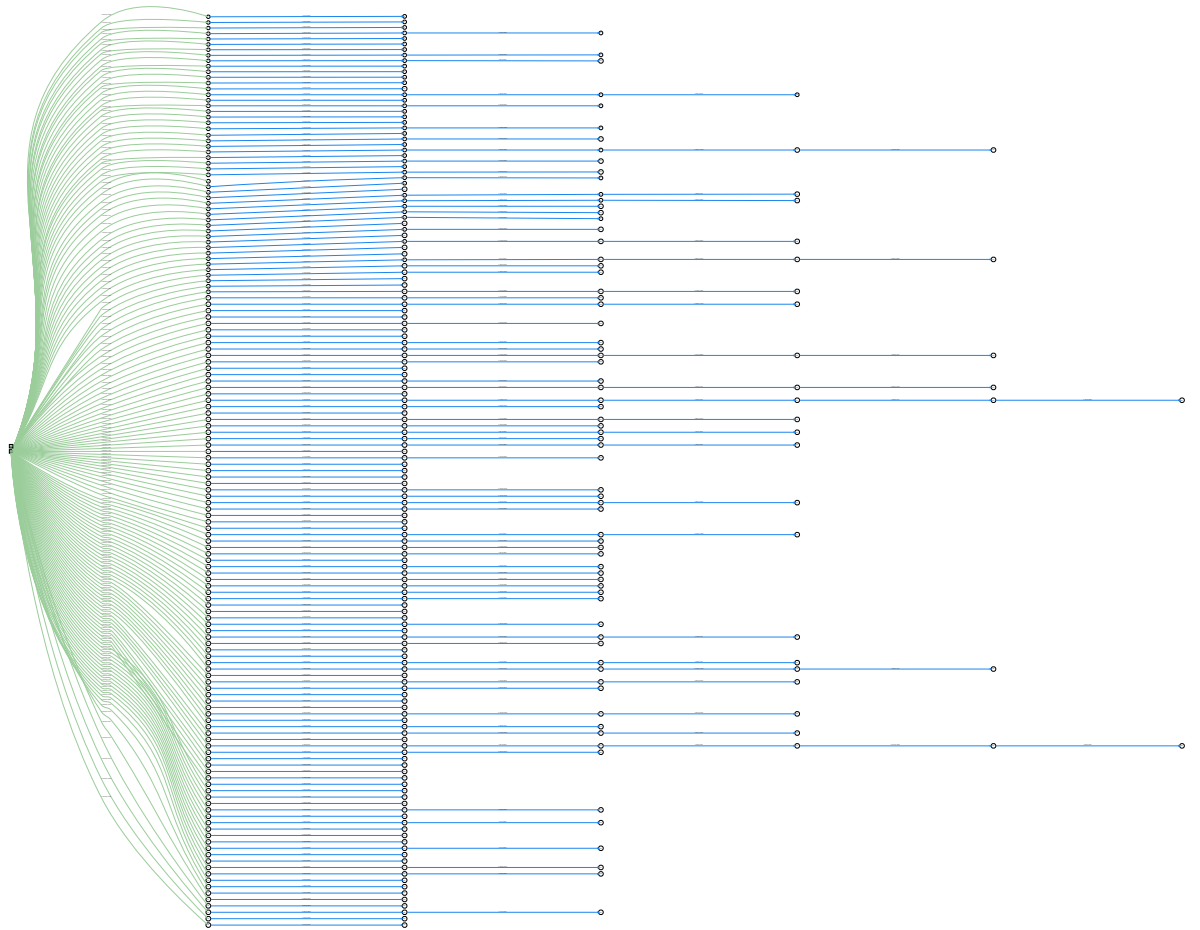


Figura A.8: BKS itinerarios *csp400*

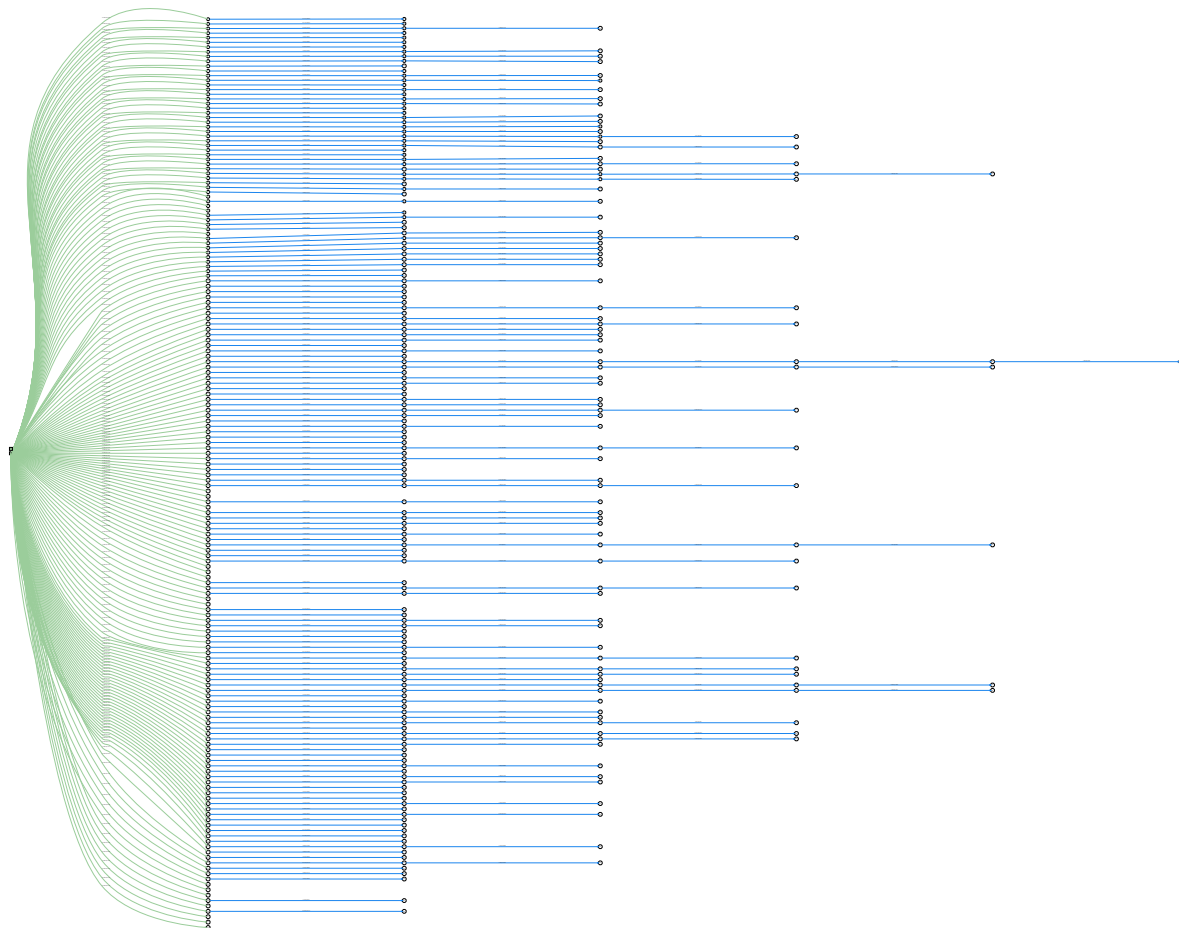


Figura A.9: BKS itinerarios *csp450*

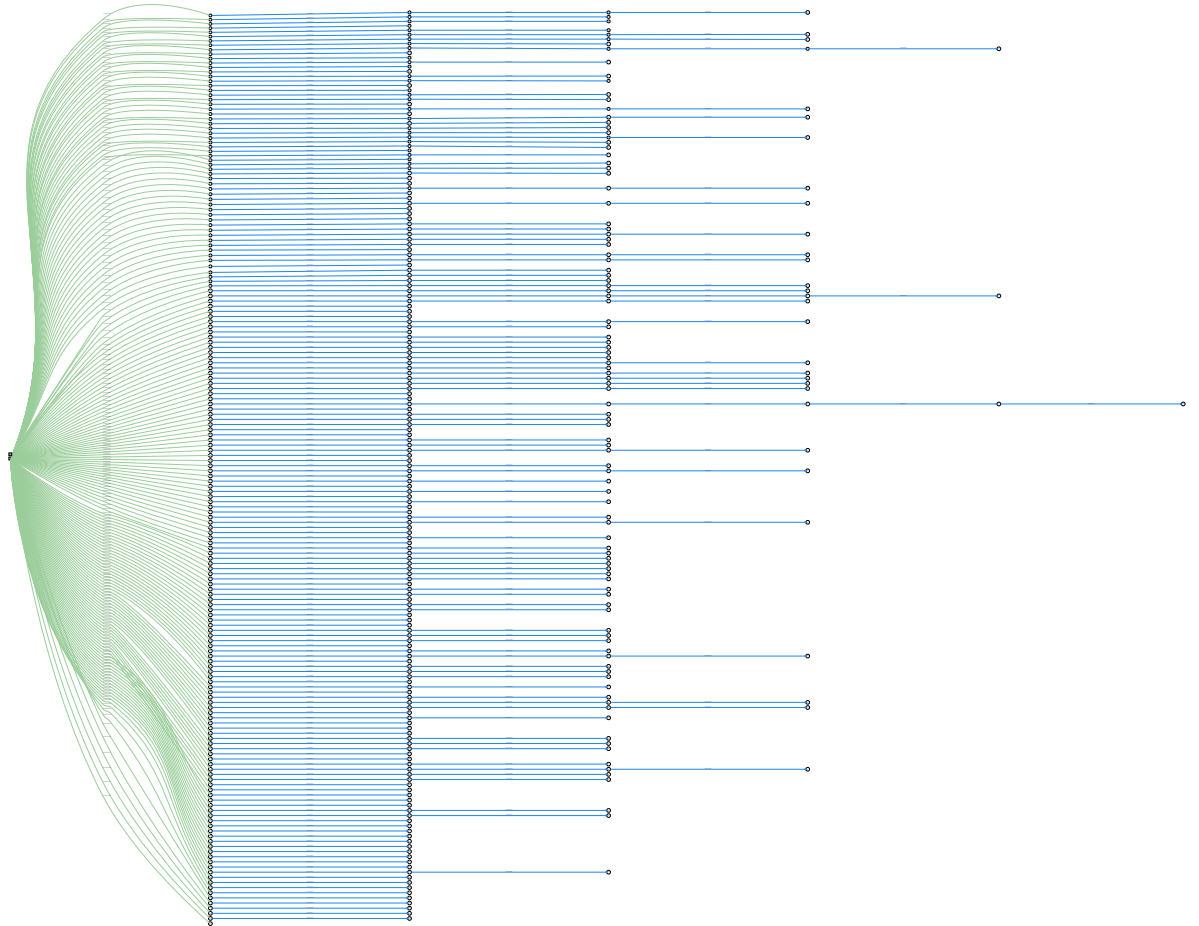


Figura A.10: BKS itinerarios *csp500*

Bibliografía

Erwin Abbink, Matteo Fischetti, Leo Kroon, Gerrit Timmer, and Michiel Vromans. Reinventing crew scheduling at netherlands railways. *Interfaces*, 35(5):393–401, 2005.

Ali Azadeh, M Hosseinabadi Farahani, H Eivazy, S Nazari-Shirkouhi, and G Asadipour. A hybrid meta-heuristic algorithm for optimization of crew scheduling. *Applied Soft Computing*, 13(1): 158–164, 2013.

John E Beasley and Nicos Christofides. An algorithm for the resource constrained shortest path problem. *Networks*, 19(4):379–394, 1989.

John E Beasley, B Cao, et al. A tree search algorithm for the crew scheduling problem. *European Journal of Operational Research*, 94(3):517–526, 1996.

Zuzana Borcinova. Two models of the capacitated vehicle routing problem. *Croatian Operational Research Review*, pages 463–469, 2017.

Marco A Boschetti, Aristide Mingozzi, and Salvatore Ricciardelli. An exact algorithm for the simplified multiple depot crew scheduling problem. *Annals of Operations Research*, 127(1-4): 177–201, 2004.

BRTData. Global brt data. <https://brtdata.org/>, 2020. Accedido en 20-06-2020.

Avishai Ceder. *Public Transit Planning and Operation: Theory, modelling and practice*. Elsevier, 1 edition, 2007.

Mingming Chen and Huimin Niu. A model for bus crew scheduling problem with multiple duty types. *Discrete Dynamics in Nature and Society*, 2012, 2012.

Shijun Chen and Yindong Shen. An improved column generation algorithm for crew scheduling problems. *Journal of Information & Computational Science*, 10(1):175–183, 2013.

Paul C Chu and John E Beasley. A genetic algorithm for the multidimensional knapsack problem. *Journal of heuristics*, 4(1):63–86, 1998.

Claudio Ciancio, Demetrio Laganà, Roberto Musmanno, and Francesco Santoro. An integrated algorithm for shift scheduling problems for local public transport companies. *Omega*, 75: 139–153, 2018.

Ulrich Derigs and Simon Schäfer. A note on extending the generic crew scheduling model of beasley and cao by deadheads and layovers. *Journal of the Operational Research Society*, 65 (5):633–644, 2014.

Guy Desaulniers and Mark D Hickman. Public transit. *Handbooks in operations research and management science*, 14:69–127, 2007.

Guy Desaulniers, Jacques Desrosiers, Marius M Solomon, François Soumis, Daniel Villeneuve, et al. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In *Fleet management and logistics*, pages 57–93. Springer, 1998.

- Martin Desrochers and François Soumis. A column generation approach to the urban transit crew scheduling problem. *Transportation science*, 23(1):1–13, 1989.
- Teresa Galvao Dias, Jorge Pinho de Sousa, and JF Cunha. Genetic algorithms for the bus driver scheduling problem: a case study. *Journal of the Operational Research Society*, 53(3):324–335, 2002.
- Dongarra, Jack J. Performance of various computers using standard linear equations software, 2009. URL <https://www.netlib.org/benchmark/performance.pdf>. [Online; accessed 10-Marzo-2021].
- Andreas T Ernst, Houyuan Jiang, Mohan Krishnamoorthy, and David Sier. Staff scheduling and rostering: A review of applications, methods and models. *European journal of operational research*, 153(1):3–27, 2004.
- Matteo Fischetti, Silvano Martello, and Paolo Toth. The fixed job schedule problem with spread-time constraints. *Operations Research*, 35(6):849–858, 1987.
- Sarah Fores, Les Proll, and Anthony Wren. Tracs ii: a hybrid ip/heuristic driver scheduling system for public transport. *Journal of the Operational Research Society*, 53(10):1093–1100, 2002.
- Manuel Fuentes, Luis Cadarso, and Ángel Marín. A new approach to crew scheduling in rapid transit networks. *Transportation Research Procedia*, 10:554–563, 2015.
- Rosmalina Hanafi and Erhan Kozan. A hybrid constructive heuristic and simulated annealing for railway crew scheduling. *Computers & Industrial Engineering*, 70:11–19, 2014.

- Dennis Huisman, Richard Freling, and Albert PM Wagelmans. Multiple-depot integrated vehicle and crew scheduling. *Transportation Science*, 39(4):491–502, 2005.
- O.J. Ibarra-Rojas, F. Delgado, R. Giesen, and J.C. Muñoz. Planning, operation, and control of bus transport systems: A literature review. *Transportation Research Part B*, 77:38–75, 2015.
- Dieter Jungnickel. *Graphs, networks and algorithms*. Springer, 2008.
- Balázs Kecskeméti and Adrián Bilics. Bus driver duty optimization using an integer programming and evolutionary hybrid algorithm. *Central European Journal of Operations Research*, 21(4): 745–755, 2013.
- Jingpeng Li and Raymond SK Kwan. A fuzzy genetic algorithm for driver scheduling. *European Journal of Operational Research*, 147(2):334–344, 2003.
- Shuguang Liu, Weilai Huang, and Huiming Ma. An effective genetic algorithm for the fleet size and mix vehicle routing problems. *Transportation Research Part E: Logistics and Transportation Review*, 45(3):434 – 445, 2009. ISSN 1366-5545. doi: <http://dx.doi.org/10.1016/j.tre.2008.10.003>. URL [//www.sciencedirect.com/science/article/pii/S1366554508001324](http://www.sciencedirect.com/science/article/pii/S1366554508001324).
- Yanping Liu, Haijun Huang, and Renyong Guo. Vehicle scheduling optimization considering the passenger waiting cost. *Journal of Advanced Transportation*, 1:1–13, 2019.
- Helena R Lourenço, José P Paixão, and Rita Portugal. Multiobjective metaheuristics for the bus driver scheduling problem. *Transportation science*, 35(3):331–343, 2001.
- César-Augusto Marín-Moreno. *Un algoritmo metaheurístico para solucionar el problema de*

- programación de la operación y labores de mantenimiento de vehículos en sistemas de transporte multidepósito*. PhD thesis, Universidad Tecnológica de Pereira, The address of the publisher, 9 2019. URL <http://repositorio.utp.edu.co/dspace/handle/11059/10886>.
- Marta Mesquita and J Paixão. Multiple depot vehicle scheduling problem: A new heuristic based on quasi-assignment algorithms. In *Computer-Aided Transit Scheduling*, pages 167–180. Springer, 1992.
- Aristide Mingozzi, Marco A Boschetti, Salvatore Ricciardelli, and Lucio Bianco. A set partitioning approach to the crew scheduling problem. *Operations Research*, 47(6):873–888, 1999.
- C Moreno, L Falcón, R Bolaños, A Subramanian, A Zuluaga, and M Echeverri. A hybrid algorithm for the multi-depot vehicle scheduling problem arising in public transportation. *International Journal of Industrial Engineering Computations*, 10(3):361–374, 2019.
- Banco Mundial. *Latinoamérica indígena en el siglo xxi: primera década*. 2015.
- Juan Carlos Muñoz and Laurel Paget-Seekins. *Restructuring public transport through Bus Rapid Transit: An intentional and interdisciplinary perspective*. Bristol, UK: Policy Press, 1 edition, 2016.
- Hande Öztop, Uğur Eliyi, Deniz Türsel Eliyi, and Levent Kandiller. A bus crew scheduling problem with eligibility constraints and time limitations. *Transportation research procedia*, 22: 222–231, 2017.
- Kenny Cárdenas Parra. *MODELO MATEMATICO LINEAL FLEXIBLE PARA LA ROTACION DE TURNOS DE TRABAJO DE LOS CONDUCTORES DEL SISTEMA DE TRANSPORTE*

MASIVO DELAREA METROPOLITANA CENTRO OCCIDENTE. PhD thesis, UNIVERSIDAD TECNOLÓGICA DE PEREIRA, 2019.

Diego Pecin, Claudio Contardo, Guy Desaulniers, and Eduardo Uchoa. New enhancements for the exact solution of the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 29(3):489–502, 2017a.

Diego Pecin, Artur Pessoa, Marcus Poggi, and Eduardo Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9(1):61–100, 2017b.

Artur Pessoa, Ruslan Sadykov, Eduardo Uchoa, and François Vanderbeck. A generic exact solver for vehicle routing and related problems. *Mathematical Programming*, 183(1):483–523, 2020.

PNUD. Programa de las naciones unidas para el desarrollo. <https://www.undp.org/content/undp/es/home/sustainable-development-goals.html>, 2020. Accedido en 20-06-2020.

Rita Portugal, Helena R Lourenço, and José P Paixão. Driver scheduling problem modelling. *Public transport*, 1(2):103–120, 2009.

Christian Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers Operations Research*, 31(12):1985 – 2002, 2004. ISSN 0305-0548. doi: [http://dx.doi.org/10.1016/S0305-0548\(03\)00158-8](http://dx.doi.org/10.1016/S0305-0548(03)00158-8). URL [//www.sciencedirect.com/science/article/pii/S0305054803001588](http://www.sciencedirect.com/science/article/pii/S0305054803001588).

Ruslan Sadykov, Eduardo Uchoa, and Artur Pessoa. A bucket graph-based labeling algorithm with application to vehicle routing. *Transportation Science*, 2020.

- Yindong Shen, Kunkun Peng, Kai Chen, and Jingpeng Li. Evolutionary crew scheduling with adaptive chromosomes. *Transportation Research Part B: Methodological*, 56:174–185, 2013.
- Barbara M Smith and Anthony Wren. A bus crew scheduling system using a set covering formulation. *Transportation Research Part A: General*, 22(2):97–108, 1988.
- Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987.
- Ingmar Steinzen, Leena Suhl, and Natalia Kliwer. Branching strategies to improve regularity of crew schedules in ex-urban public transit. *OR spectrum*, 31(4):727, 2009.
- Yi Su and Gunnar Lucko. Linear scheduling with multiple crews based on line-of-balance and productivity scheduling method with singularity functions. *Automation in Construction*, 70: 38–50, 2016.
- Anand Subramanian, Eduardo Uchoa, and Luiz Satoru Ochi. A hybrid algorithm for a class of vehicle routing problems. *Computers & Operations Research*, 40(10):2519–2531, 2013.
- Satoru Takahashi, Kenji Kataoka, Teruhito Kojima, and Masayuki Asami. An algorithm for automatically modifying train crew schedule. *IEEEJ Transactions on Industry Applications*, 128: 1291–1297, 2008.
- Zhiqiang Tian and Qi Songa. Modeling and algorithms of the crew scheduling problem on high-speed railway lines. *Procedia-Social and Behavioral Sciences*, 96:1443–1452, 2013.
- Attila Tóth and Miklós Krész. An efficient solution approach for real-world driver scheduling

problems in urban bus transportation. *Central European Journal of Operations Research*, 21(1): 75–94, 2013.

Jorne Van den Bergh, Jeroen Beliën, Philippe De Bruecker, Erik Demeulemeester, and Liesje De Boeck. Personnel scheduling: A literature review. *European journal of operational research*, 226(3):367–385, 2013.

Joyce W Yen and John R Birge. A stochastic programming approach to the airline crew scheduling problem. *Transportation Science*, 40(1):3–14, 2006.

Gang Yu and Xiangtong Qi. *Disruption management: framework, models and applications*. World Scientific, 2004.

Liping Zhao. A heuristic method for analyzing driver scheduling problem. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 36(3):521–531, 2006.