



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

ALGORITMO GENÉTICO PARA EL DESPACHO DE TÉCNICOS CON
DEPENDENCIA DEL TIEMPO

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INDUSTRIAL

JOSÉ IGNACIO BRAVO VALENZUELA

PROFESOR GUÍA:
RICARDO SAN MARTÍN ZURITA

MIEMBROS DE LA COMISIÓN:
RODOLFO URRUTIA URIBE
CHARLES THRAVES CORTES-MONROY

SANTIAGO DE CHILE
2017

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INDUSTRIAL
POR: JOSÉ IGNACIO BRAVO VALENZUELA
FECHA: 2/10/2017
PROF. GUÍA: RICARDO SAN MARTÍN ZURITA

ALGORITMO GENÉTICO PARA EL DESPACHO DE TÉCNICOS CON DEPENDENCIA DEL TIEMPO

La empresa Xerox en Chile ofrece el servicio de reparación de equipos de oficina tales como impresoras y fotocopiadoras. Para ello cuenta con una dotación de técnicos que se trasladan hacia los clientes para realizar el trabajo. La estrategia de la empresa es atender las solicitudes de trabajo tan pronto como sean recibidas, es decir, organizar y atender durante el día las llamadas que van recibiendo por parte de sus clientes. Cuando esto no es posible, ciertos clientes son postergados para otro día. Como consecuencia de dicha estrategia, las llamadas que no alcanzaron a ser atendidas a tiempo por los técnicos llegó al 54% para el periodo 2002-2003 en la ciudad de Santiago, considerándose esto un deterioro en la calidad de atención.

En este trabajo se propuso disminuir la cantidad de clientes postergados junto con la distancia recorrida mediante la planificación eficiente de las rutas. La metodología consistió en abordar la situación como un problema de ruteo de vehículos, donde además los tiempos de viaje no son fijos, sino que dependen de la hora del día. Este problema es conocido como el Problema de Ruteo de Vehículos con Dependencia del Tiempo (TDVRP por sus siglas en inglés). Debido a la necesidad de encontrar soluciones rápidas, el problema fue resuelto mediante Algoritmo Genético, el cual consiste en generar un conjunto de rutas aleatorias de clientes y luego intercambiar entre sí pequeños segmentos para crear nuevas y mejores rutas, imitando de tal forma el proceso de recombinación genética presente en la naturaleza. Se consideró también un cambio en la política de atención, donde ahora todos los llamados serían atendidos al día siguiente para una mejor planificación.

Los resultados se obtuvieron para una semana de operación. Las soluciones alcanzaron tasas de retraso entre 6-20% en comparación con el 59% de la situación real de aquella semana. Respecto de la dependencia del tiempo, los resultados mostraron que no hubo diferencias significativas al incluir variación en los tiempos de viaje en comparación con tiempos de viaje fijos.

Se concluye que el cambio en la estrategia de atención y una buena planificación inicial mejoran ostensiblemente la cantidad de clientes que deben ser postergados, esto implica una oportunidad de mejora en la calidad de servicio que ofrece Xerox. Por otro lado, incluir dependencia de la hora del día en los tiempos de viaje no significó un aporte al modelo en cuanto calidad de las soluciones. Aquello podría explicarse por tiempos de servicio mayores que los tiempos de viaje, haciendo que el impacto de estos últimos sean irrelevantes para la solución del modelo en una ciudad.

Tabla de Contenido

1. Introducción	1
1.1. Antecedentes	1
1.2. Justificación	2
1.3. Objetivos	2
1.3.1. Objetivo General	2
1.3.2. Objetivos Específicos	2
1.4. Alcances	3
2. Revisión Bibliográfica	4
2.1. Problema de ruteo de vehículos con dependencia del tiempo (TDVRP)	4
2.2. Generación de Columnas	6
2.3. Trabajos anteriores	7
3. Definición del problema	8
3.1. Despacho de técnicos	8
3.2. Matriz de tiempos de viaje	13
3.2.1. Open Street Map	14
3.2.2. Velocidad de buses Transantiago	14
3.2.3. Estimación del tiempo de viaje	15
3.3. Modelo matemático	16
3.4. Valor de penalización	19
3.4.1. Clasificación de clientes	19
3.4.2. Tiempo de viaje esperado	19
3.4.3. Competitividad del mercado	20
3.4.4. Tasa de fuga	20
4. Heurística	22
4.1. Idea general	22
4.2. Generación de columnas	24
4.3. Algoritmo Genético	25
5. Resultados	29
5.1. <i>Benchmark</i>	29
5.2. Resultados computacionales	30
6. Conclusión	37

Capítulo 1

Introducción

1.1. Antecedentes

El problema de ruteo de vehículos (*vehicle routing problem* o VRP) consiste en encontrar la ruta que minimice el tiempo de viaje. Por lo general es abordado desde la programación lineal, donde además existen muchas variantes de problema original, tales como ventanas de tiempo para atender los clientes o límites de capacidad de los vehículos.

Una aplicación del VRP se puede identificar en el trabajo que realiza la empresa Xerox en Chile. Su operación diaria enfrentan el problema de despacho de técnicos, que consiste en las siguientes características:

- Se tiene una cantidad fija de técnicos que deben atender solicitudes de reparación dentro de una zona geográfica.
- Los técnicos deben desplazarse durante el día para atender todos los clientes.
- Las solicitudes de reparación deben cumplirse lo más pronto posible.

El Departamento de Ingeniería Industrial posee la base de datos real del negocio para el periodo 2002-2003, contando con información detallada de los tiempos de traslado y servicio para cada cliente. Aquello ha permitido plantear y solucionar distintas variantes del problema original:

- Diseño de flota con horizonte de un año para demanda estocástica[1].
- Inserción dinámica de clientes con ventanas de tiempo[2]
- Ruta óptima para tiempos de servicio estocásticos[3].
- Ruta óptima para ventanas de tiempo suaves de un día[4].
- Solución heurística para ruteo con predicción de demanda estocástica[5].

Según lo expuesto, ya han sido desarrollado modelos para abordar las características de demanda estocástica y diseño de flota, pero no existe hasta el momento uno que considere tiempos de viaje variables según la hora del día. A este nuevo tipo de problema se le conoce

como ruteo de vehículos con dependencia del tiempo (*time dependent vehicle routing problem* o TDVRP). Por lo tanto es de interés formular el problema considerando tal punto y estudiar cómo afecta a la planificación de las rutas.

1.2. Justificación

El objetivo del trabajo es formular el modelo de programación lineal del TDVRP aplicado al problema de despacho de técnicos, el cual incluye la decisión de aplazar la atención de los clientes. Según la revisión hecha por Gendreau, Ghiani & Guerriero [6], introducir dependencia en los tiempos de viaje logra reducir el tiempo total de viaje para el ruteo de vehículos, por lo tanto se espera que esta variante mejore los tiempos de despacho de la planificación, con el consiguiente aumento de clientes atendidos durante el día. La novedad del trabajo, a diferencia de otros previos sobre el TDVRP, es la utilización de datos reales sobre la matriz origen-destino de la ciudad, haciendo posible cuantificar el verdadero beneficio de incluir dependencia del tiempo para la planificación de la empresa. Por ese motivo, la información de Xerox se complementará con la proporcionada por el proyecto OpenStreetMap y Transantiago.

Un aspecto relevante sobre el TDVRP es la definición de los tiempos de viaje entre nodos. Lo usual es dividir el día en periodos, asignándole a cada uno un tiempo de viaje distinto. Como la medida es arbitraria, se pueden establecer tantos intervalos como se quiera: desde periodos de 10 minutos hasta 1 hora. En los trabajos que emplean datos ficticios no puede concluirse cuál es el intervalo adecuado, dado que no representan el tráfico real de la ciudad. Por este motivo, aprovechando la disponibilidad de la información de Xerox y OpenStreetMap, se realizará un análisis de sensibilidad sobre la matriz origen-destino y de ese modo podrán darse orientaciones acerca del intervalo adecuado para la ciudad de Santiago.

1.3. Objetivos

1.3.1. Objetivo General

Diseñar una heurística que permita resolver el problema de despacho de técnicos con tiempos de viaje dependientes del tiempo.

1.3.2. Objetivos Específicos

- Documentar los modelos actuales para el TDVRP.
- Construir la matriz origen-destino del sector sur-oriente de Santiago según hora del día.
- Desarrollar el modelo de programación lineal que represente el problema de despacho de técnicos con tiempos de viaje dependientes del tiempo.

- Proponer una heurística que encuentre una solución aceptable en menos de 30 minutos para un día de operación.
- Resolver el problema utilizando la base de datos de Xerox y realizar un análisis de sensibilidad sobre los periodos de la matriz origen-destino.

1.4. Alcances

El alcance del trabajo es proponer una planificación de ruteo para el problema de despacho de técnicos que enfrenta la empresa Xerox. La solución propuesta se limitará al sector sur-oriente de Santiago de Chile, utilizando la base de datos de la empresa para el periodo 2002-2003. Se establecerá que la cantidad de técnicos es fija, esto quiere decir que se buscará mejorar la distribución de los técnicos. También se considerará que los tiempos de servicio son deterministas y que la demanda se conoce el día anterior, de forma que no existe aleatoriedad en tales aspectos.

Por otro lado, la base de datos presenta una limitante. Si bien su información pertenece a los años 2002-2003, la estimación del tiempo de viaje para la matriz origen-destino se realizará con información actual. Por consiguiente implica tomar como supuesto que el tráfico vehicular es el mismo hace 15 en el Gran Santiago y que las calles y avenidas se han mantenido intactas, tanto en cantidad como en su sentido del tránsito.

Capítulo 2

Revisión Bibliográfica

2.1. Problema de ruteo de vehículos con dependencia del tiempo (TDVRP)

El problema de ruteo de vehículos con dependencia del tiempo (TDVRP por sus siglas en inglés) ha sido definido por Malandraki & Daskin[7] de la siguiente manera:

- Una flota de vehículos de capacidad fija debe atender clientes de demanda fija desde una bodega central.
- Los clientes son asignados a cada vehículo y su ruta, de manera que el tiempo total de las rutas sea minimizado.
- El tiempo de viaje, sea entre dos clientes o la bodega, depende de la distancia a recorrer y del horario del día.

La condición de los tiempos de viaje estocásticos según la hora del día es tratado por primera vez en 1989 por Malandraki[8], y luego en 1992 por Malandraki & Daskin[7]. En este último desarrollan una formulación de programación lineal donde los tiempos de viaje son representados por una función escalonada, cuyos valores son función del horario del día, es decir, el tiempo de viaje para un mismo tramo no es el mismo si el traslado se produce en la mañana, a mediodía o en la tarde. Aunque el trabajo se enfoca en el problema genérico de ruteo de vehículos, también plantea una solución para incluir ventanas de tiempo en las visitas de los clientes.

Gendreau, Ghiani & Guerriero[6] realizan una revisión completa en la literatura sobre dependencia del tiempo en el ruteo, exponiendo las ventajas de incorporar esta consideración a los modelos de programación lineal de VRP. Entre algunos de los resultados cuantitativos encontrados por la revisión, considerar dependencia en el tiempo reduce entre 22-25 % el consumo de bencina, reduce un 7 % la emisión de gases contaminantes y reduce la mercancía perecible estropeada en 47 %. Otras de las ventajas son la reducción de los tiempos de viaje[9] y la reducción del tamaño de la flota de vehículos[10].

Por otro lado, la principal desventaja de los estudios reseñados es que todos prueban sus soluciones con las instancias propuestas por Marius Solomon¹. Ésta es una base de datos ficticia compuesta de 6 conjuntos, cada uno de 100 clientes distribuidos espacialmente al azar. Las instancias se clasifican en los tipos R (distribuidos aleatoriamente), C (distribuidos en *clusters*) y RC (mezcla entre los tipos R y C). Se recurre a estos datos porque es complicado conseguir una matriz de tiempos de viaje origen-destino calculada con información real. Además las instancias de Solomon contienen únicamente distancias entre nodos, por lo cual cada trabajo debió generar sus propios tiempos de viaje, impidiendo comparar soluciones. En general los tiempos de resolución son menores a 5 minutos para 25-100 clientes. Asimismo resulta una desventaja que los modelos propuestos están basados en heurísticas, lo que impide apreciar sus beneficios en comparación a calcular la solución exacta.

Algunas de las heurísticas que se han empleado para resolver el TDVRP son las siguientes:

- Algoritmo genético: Heurística basada en la estrategia de combinación genética para mejorar la función objetivo. Para el VRP se formula recombinar tramos de las rutas y dejar como sobrevivientes las rutas que entreguen el menor tiempo de viaje total. Ejemplo es el trabajo de Kumar & Panneerselvam[10].
- Búsqueda tabú: Heurística basada en la idea de encontrar óptimos dentro de un vecindario (*neighborhood*) en iteraciones sucesivas y recordar los mejores óptimos locales. Ejemplos de este método son el intercambio de arcos en las rutas, como 2-opt y Or-opt. Ejemplo de esto es el trabajo de Taillard *et al*[11].

El problema de despacho de técnicos, en su versión de tiempos deterministas, ha mostrado tiempos de resolución exacto entre 1-3 horas para 78 clientes[4]. Si bien es una desventaja con respecto a los tiempos de resolución mostrados por las heurísticas, deben ser considerados dos puntos: I) no existen trabajos que comparen soluciones exactas con heurísticas para tiempos estocásticos de viaje y dependientes del tiempo, todo bajo una misma base de datos; II) el despacho de técnicos considera demanda determinista, conocida el día anterior. En consecuencia resulta interesante obtener una instancia de datos reales sobre la que pueda trabajarse y comparar soluciones exactas contra heurísticas, para finalmente cuantificar los beneficios/costos de cada uno en términos de tiempo de resolución y función objetivo.

Debido a la inexistencia de datos reales en los problemas de dependencia en el tiempo, no existen fórmulas para calcular sus parámetros asociados. En la revisión de Gendreau, Ghiani & Guerriero[6] todos los trabajos utilizan datos simulados. El parámetro que cambia durante el día es la velocidad del vehículo, así calculan los tiempos de viaje dividiendo la distancia entre nodos por la velocidad promedio. En el estudio realizado por Cortés *et al*[4] calculan los tiempos de viaje como el tiempo promedio intracomunal para los viajes dentro de la comuna y tiempo promedio intercomunal para los viajes fuera de la comuna. Hill & Benton[12] han abordado el problema de estimar el parámetro de tiempo en el contexto de Investigación de Operaciones. Proponen calcular las velocidades del vehículo en función de las vecindades de cada nodo, aprovechando de ese modo la información disponible sobre viajes pasados. Primero se plantea la fórmula para estimar la velocidad entre los nodos i y j en el periodo T como $r_{ijT} = (r_{iT} + r_{jT}) / 2$, luego la distancia entre i y j es dividida por la velocidad r_{ijT} , obteniendo como resultado la velocidad para el periodo T . La ventaja de esta

¹<http://w.cba.neu.edu/~msolomon/problems.htm>

formulación es que no requiere que todos los viajes se hayan realizado alguna vez, basta con tener unos pocos para estimar el resto de los viajes cuya información histórica no existe. Por otro lado, su debilidad es que requiere que los viajes hayan sucedido en el periodo estudiado. Por ejemplo, si se desea estimar el tiempo de viaje entre i y j en el periodo T , al menos debe existir información de un viaje que involucre a i y otro a j durante ese periodo.

2.2. Generación de Columnas

Cuando la cantidad de variables es mayor que la cantidad de restricciones en un problema de programación lineal, muchas de las variables tendrán valor cero en la solución óptima o factible debido a que no serán incluidas. La técnica de Generación de Columna (GC) consiste en separar el problema de forma que solamente se generen columnas (patrones) cuando sean necesarias para mejorar la solución, es decir, se buscan las variables que puedan mejorar la función objetivo, ya sea minimizando o maximizando según corresponda. Este esquema es útil cuando el problema es grande en términos de variables y no puede ser resuelto en tiempo factible o la cantidad de memoria computacional requerida es excedida por las características de la máquina. Lübbecke y Desrosiers[13] dan una descripción formal del método y sus aplicaciones a otros tipos de problemas: problema del vendedor viaje, *scheduling*, asignación y *set partitioning*. Una explicación más detallada también puede encontrarse en *Introduction to Linear Optimization* de Bertsimas y Tsitsiklis[14].

El esquema usual en Generación de Columnas consiste en dos problemas de programación lineal: un Problema Maestro y un Subproblema asociado. El Problema Maestro representa el problema de optimización propiamente tal a resolver, mientras que el Subproblema se encarga de encontrar un patrón con costo reducido negativo/positivo, con el propósito de disminuir/aumentar el valor actual de la función objetivo. En una iteración típica primero se resuelve el problema maestro y se obtienen las variables duales, luego se utilizan estas variables como coeficientes para formular el subproblema. Al resolver el subproblema se obtiene, si existe, una nueva columna que mejora la solución del Problema Maestro, siendo introducida en este último (ver figura 2.1).

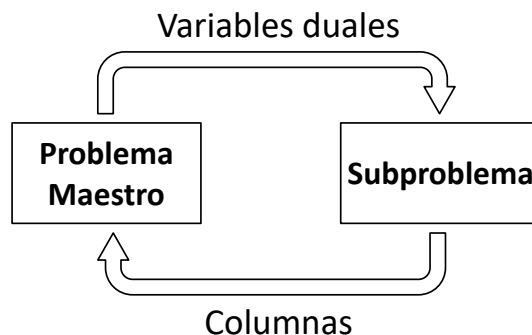


Figura 2.1: Método de resolución en la Generación de Columnas.

2.3. Trabajos anteriores

La base de datos de Xerox, debido a la información detallada que posee respecto a los tiempos de servicio y traslado, ha sido utilizada en varios trabajos relacionados con ruteo de vehículos. Cortés *et al.* en 2011[1] emplearon métodos estadísticos para simular la demanda semanal de llamadas, con eso estimaron la cantidad de técnicos necesarios para luego proponer una política de contratación de técnicos, con horizonte de planificación a un año. Rojas en 2012[2] resuelve el problema dinámico de ruteo mediante una heurística que involucra la técnica de Generación de Columnas. Su modelo utiliza la información histórica para generar puntos de espera donde se concentrarán las llamadas futuras. Hecho el ruteo inicial, se va modificando a medida que llegan nuevos llamados, lo cuales son insertados en las rutas existentes de forma óptima. Souyris *et al.* en 2013[3] proponen un modelo de optimización robusta para tiempos de servicio estocásticos, siendo éstos muy variables en la práctica. Para resolverlo recurren al esquema de *Branch-and-Price*, que también involucra Generación de Columnas. Briceño en 2014[4] vuelve a resolver el problema dinámico de ruteo, pero considerando ahora una predicción de la demanda basada en información histórica. Con esa predicción define los puntos de espera, donde es más probable que aparezcan las llamadas, insertándolas en las rutas actuales. Emplea una heurística que aprovecha la Generación de Columnas. Por último, Cortés *et al.* en 2014[4] resuelve el problema exacto utilizando las técnicas de *Branch-and-Price* y Programación por Restricciones (*Constraint Programming*). Como instancia de prueba se resuelve un solo día.

Capítulo 3

Definición del problema

3.1. Despacho de técnicos

El presente trabajo está basado en el problema real que enfrenta la empresa Xerox. Su labor consiste en ofrecer el servicio de reparación de equipos para oficina: impresoras y fotocopadoras. Su servicio tiene cobertura a nivel nacional, por eso cuenta con una dotación de técnicos asignados a diferentes zonas geográficas del país. Atiende los días laborales de lunes a viernes.

Cuando un cliente llama a Xerox para solicitar una reparación, se despacha un técnico hasta su ubicación, luego procede a reparar el equipo que presenta fallas. La atención del cliente no es inmediata, sino que depende de la disponibilidad de técnicos en el momento. Bajo ese escenario, un cliente puede ser atendido apenas realiza la llamada o después de esperar algunas horas o incluso ser atendido días posteriores. El orden de atención durante el día queda a decisión de Xerox, por esa razón cobra importancia una buena planificación del ruteo; permitiría aumentar la cantidad de clientes atendidos durante el día, con la consiguiente disminución de clientes postergados, así como también disminuir el tiempo de traslado de los técnicos. El objetivo de Xerox será entonces minimizar simultáneamente la cantidad de llamadas que deben ser postergadas para otro día y el tiempo de viaje de los despachos.

Tanto la heurística como la programación lineal que resuelven la planificación del ruteo (capítulos 4 y 5) serán probados con los datos de las llamadas telefónicas recibidas por Xerox durante el periodo 2002-2003. La zona de estudio se elige basándose en las elecciones hechas en los trabajos previos relacionados con la misma base de datos. El motivo es generar resultados comparables entre estudios. Cortés *et al.*[1, 4], Rojas[2] y Briceño[5] fijan su atención en el sector Sur-Oriente de Santiago, dando como argumentación la baja densidad de llamas por superficie que posee dicho territorio de la capital, haciendo relevante una buena planificación de las rutas. Por lo tanto se decide situar la zona de estudio en ese sector. La figura 3.1 muestra las comunas de Santiago que fueron seleccionadas para el estudio, mientras que el cuadro 3.1 detalla la distribución de las llamadas. La mayoría de las llamadas se concentra en la zona urbanizada de la ciudad, representada por el sombreado gris.

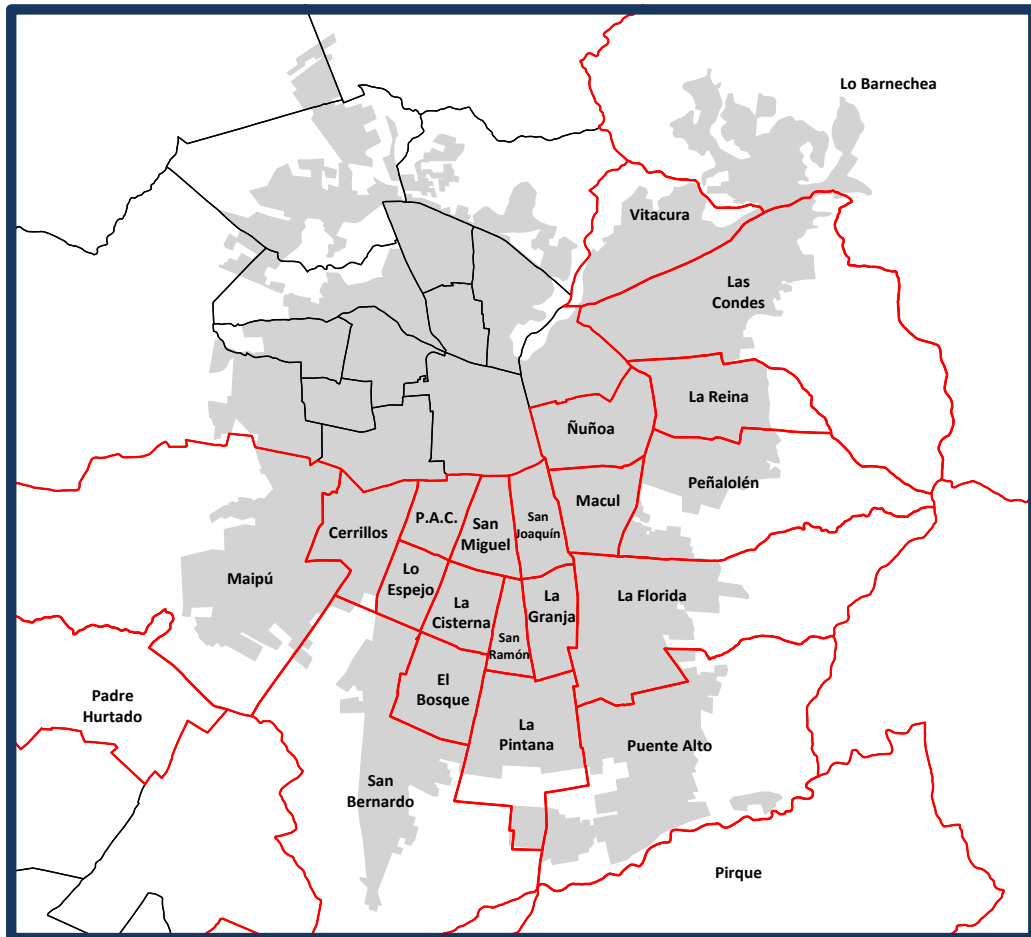


Figura 3.1: División administrativa del Gran Santiago. En gris se demarca la zona urbanizada de la ciudad, mientras que las comunas seleccionadas para la zona de estudio son etiquetadas con su nombre y delimitadas con rojo.

Comuna	Llamadas	Porcentaje
Las Condes	11.692	36,0
Ñuñoa	3.795	11,7
Macul	2.551	7,9
Vitacura	2.343	7,2
Maipú	1.553	4,8
San Bernardo	1.551	4,8
La Florida	1.518	4,7
Cerrillos	1.240	3,8
San Miguel	1.112	3,4
Puente Alto	1.046	3,2
La Reina	834	2,6
San Joaquín	695	2,1
Lo Barnechea	658	2,0
El Bosque	361	1,1
Peñalolén	349	1,1
La Cisterna	298	0,9
Padre Hurtado	264	0,8
San Ramon	123	0,4
La Pintana	113	0,3
La Granja	107	0,3
Pedro Aguirre Cerda	97	0,3
Lo Espejo	75	0,2
Pirque	59	0,2
Total	32.434	

Cuadro 3.1: Cantidad de llamadas por comuna dentro de la zona de estudio.

Mes	Llamadas por día	Mes	Llamadas por día
Enero	114	Julio	147
Febrero	116	Agosto	142
Marzo	148	Septiembre	134
Abril	133	Octubre	138
Mayo	156	Noviembre	135
Junio	141	Diciembre	113

Cuadro 3.2: Promedio de llamadas diarias para la zona de estudio, según mes del año.

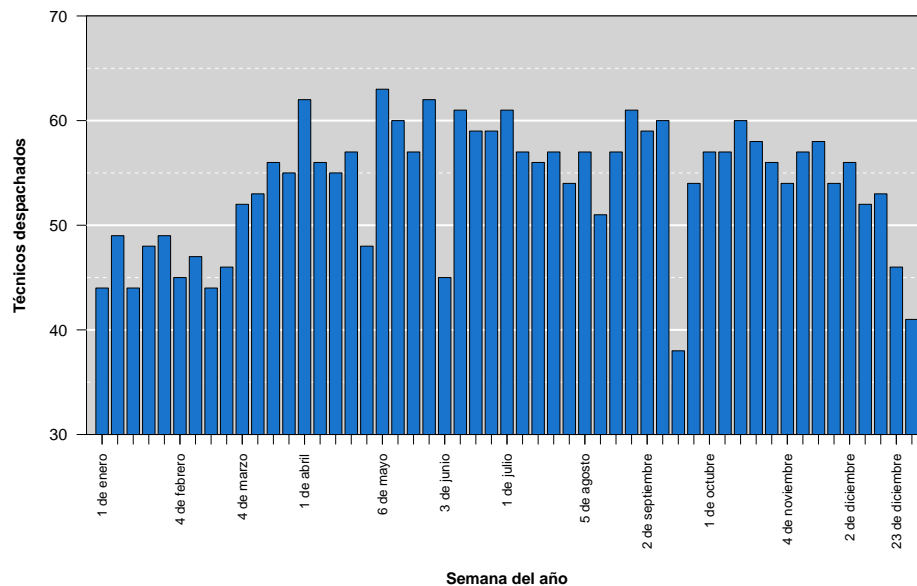


Figura 3.2: Cantidad de técnicos despachados según semana del año.

La cantidad diaria de clientes que solicitan una reparación ronda en 140 promedio. El cuadro 3.2 detalla los promedio diarios por mes. Se observa que diciembre, enero y febrero son los meses con menor actividad, lo que es previsible dado que corresponde al periodo de vacaciones. El mes de mayor actividad es mayo con 156 llamadas promedio por día. Respecto a la cantidad de técnicos, Xerox no emplea un número fijo, sino que es variable. La figura 3.2 calcula el promedio diario de técnicos que trabajaron según semana. Como era de esperar, las semanas correspondientes al mes de mayo fueron las que despacharon mayor cantidad. En general se dispone de 40 a 60 técnicos por día durante el año.

La estrategia actual de Xerox para la operación diaria es atender los llamados tan pronto sean recibidos, de ser posible el mismo día. La figura 3.3 constata que poco más de la mitad de clientes suelen ser atendidos durante el día, mientras que los restantes son atendidos al día siguiente. Solamente una fracción son atendidos después de 2 días o más.

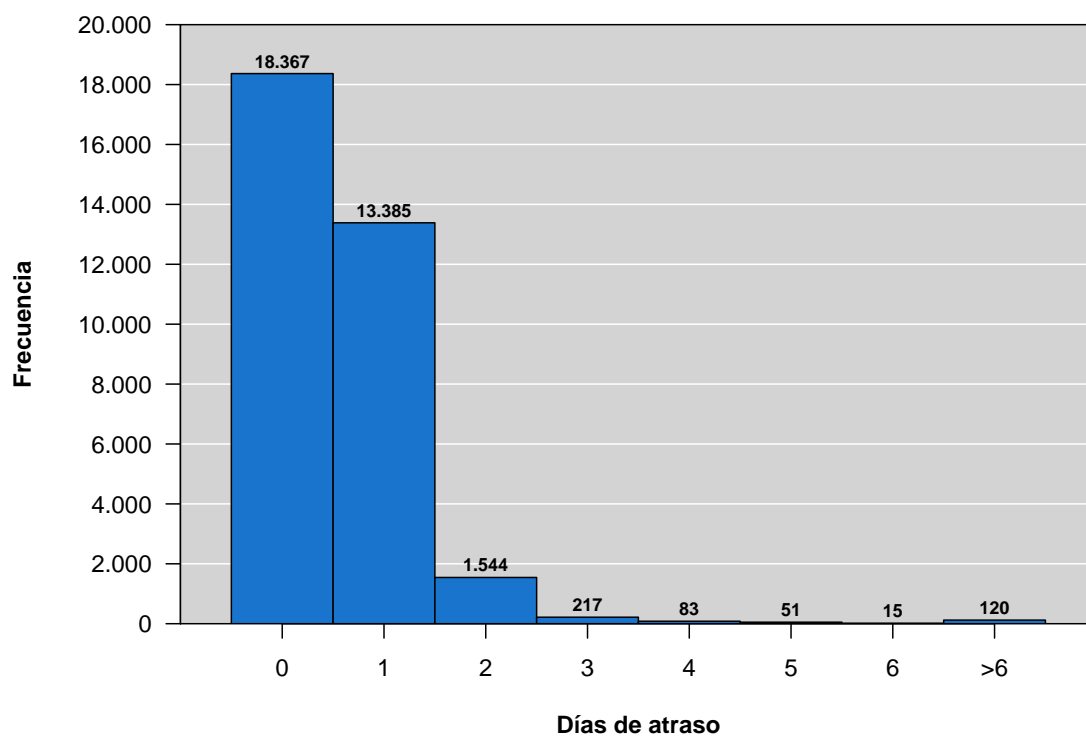


Figura 3.3: Frecuencia de atrasos en la atención de clientes. El valor 0 indica que el cliente fue atendido el mismo día que la llamada fue efectuada.

Familia	μ	σ	Familia	μ	σ
CGS	63	58	ODP-CNV L&L	85	53
CH-COP. PERSONAL	64	55	ODP-COLOR	137	95
CH-FAX	34	31	ODP-DCS DPTL	113	67
CH-INKJET	28	31	ODP-DCS WG	89	53
CH-LASER	51	40	ODP-DTL	128	80
H-LASER COLOR	68	53	ODP-PRO	52	39
CH-MULTIF.	40	38	ODP-WG	108	62
DPS	140	104	PS	160	92
ODP-CNV DIG	81	52	XES	134	94

Cuadro 3.3: Tiempos de servicio para distintas familias de equipos. Promedios (μ) y desviaciones estándar (σ) en minutos.

Con el fin de facilitar el modelamiento matemático del problema, la operación diaria teórica supondrá características distintas de la situación real expuesta anteriormente:

- Las llamadas de los clientes son clasificadas de acuerdo al tipo de problema. La base de datos cuenta con información sobre el tiempo de servicio efectivo de cada llamada, lo que servirá para calcular el tiempo promedio por tipo de servicio (ver cuadro 3.3). Se asume que ese tiempo promedio efectivamente se realiza cuando llega el técnico donde el cliente.
- Las llamadas son recibidas y organizadas para ser atendidas al día hábil siguiente. Por lo tanto puede elegirse de antemano no visitar un cliente y postergarlo para el día siguiente, aunque será obligatorio atenderlo en ese día. Esto implica que un cliente puede tener como máximo un día de atraso en la atención.
- Los técnicos comienzan el día en la posición de algún cliente. Esto implica que la ruta no comienza ni finaliza en una terminal o *depot*.
- La cantidad de técnicos disponibles es fija durante todo el año.
- El horario de trabajo es de 7:00 a 18:30 horas. El último trabajo debe terminar antes de las 18:30 horas.

3.2. Matriz de tiempos de viaje

Los tiempos de viaje entre clientes son estimados usando en conjunto dos bases de datos. La primera contiene las distancias de viaje dentro de la ciudad, basadas en el proyecto Open Street Map. La segunda contiene las velocidades según la hora del día, reconstruidas a partir de las velocidades de los buses del sistema de transporte público Transantiago.

Aunque existen servicios que pueden entregar la información sobre tiempos de viaje para un conjunto de pares origen-destino, como la API de Google Maps, estos son muy costosos. Por tal motivo se elige utilizar información gratuita disponible para cualquier persona.

3.2.1. Open Street Map

El proyecto OpenStreetMap¹ es un repositorio de mapas que entrega información detallada sobre calles y carreteras de todo el mundo, aunque también posee información relativa a señalética, semáforos y numeración de inmuebles, entre otros. Es un proyecto colaborativo mantenido y actualizado por personas voluntarias, cuya calidad de los mapas fue estudiada por Mordechai Haklay[15] para el caso de Inglaterra y la ciudad de Londres. Determinó, comparando con datos oficiales sobre información geográfica, que la precisión de los objetos estaba en promedio dentro de los 6 metros y que las rutas coincidían en un 80 %, concluyendo que el mapa colaborativo era bastante preciso.

Para definir las rutas entre los distintos clientes en el problema de despacho, se utilizó el mapa de OSM correspondiente a la ciudad de Santiago. El criterio para calcular las rutas fue minimizar la distancia recorrida. Su cálculo se realizó usando el motor de ruteo Open Source Routing Machine[16] (OSRM por sus siglas), código también colaborativo que permite procesar los mapas de OpenStreetMap y realizar operaciones de ruteo.

Aunque el criterio establecido aquí para definir la ruta entre dos direcciones es la que minimice la distancia recorrida, esto no significa necesariamente que también minimice el tiempo de viaje. Puede ocurrir que una ruta minimice la distancia recorrida, no obstante si se ve afectado por la congestión vehicular, el tiempo de viaje podría resultar mayor en comparación a caminos alternativos más largos pero menos congestionados.

3.2.2. Velocidad de buses Transantiago

Una vez obtenida las distancias entre direcciones, se puede estimar su tiempo de viaje mediante la velocidad de conducción. Eventualmente es difícil obtener dichos datos considerando cada ruta y hora del día particular. Una solución es utilizar la información de las velocidades que registran los buses del sistema de transporte público Transantiago. Ésta se encuentra disponible en la página web del Directorio de Transporte Público Metropolitano², donde las mediciones corresponden al año 2010. En el cuadro 3.4 se expone un extracto de la información que puede ser obtenida. La base de datos registra las velocidades cada media hora, de 00:00 a 23:30, para días laborales y fines de semana.

Las mediciones evidencian que la velocidad no es constante en el tiempo, como era de esperar. La figura 3.4 muestra que la velocidad de conducción disminuye alrededor de las 7:30 y 18:00 horas, horario que corresponde al inicio y término de la jornada laboral, respectivamente. Por otro lado, las velocidades aumentan a las 9:00 y 14:00 horas. La línea roja

¹<http://www.openstreetmap.org>

²<http://www.dtpm.cl/index.php/estadisticas> [consulta: 25 mayo 2017]

Servicio	Sentido	Velocidad cada media hora [Km/hr]						
		07:00	07:30	08:00	08:30	09:00	09:30	10:00
D03	Ida	18,39	16,02	15,92	16,82	17,30	17,01	16,70
506	Regreso	20,80	20,97	21,69	22,27	22,36	21,76	21,27

Servicio	Sentido	Velocidad cada media hora [Km/hr]						
		15:00	15:30	16:00	16:30	17:00	17:30	18:00
D03	Ida	16,57	16,07	15,70	15,48	15,06	14,26	13,40
506	Regreso	20,88	20,61	20,11	19,43	18,90	18,49	18,20

Cuadro 3.4: Ejemplo de la información proporcionada por el DTPM para el servicio de buses del Transantiago.

indica la velocidad promedio de 22,1 km/h, valor que corresponde a las 13:00 horas. Este horario será la referencia para modelar una instancia donde no existe dependencia del tiempo al momento de calcular la matriz de distancia.

3.2.3. Estimación del tiempo de viaje

Las velocidades del transporte público son distintas a las que alcanzan los vehículos particulares, las causas son las paradas que debe hacer un bus para recoger pasajeros y el recorrido fijo que debe seguir. La estimación obtenida de Transantiago debe ser corregida sumando 5, 8 km/h a las velocidades promedios de los buses. El factor de corrección se justifica porque es la diferencia de velocidad promedio entre el transporte público y privado³.

Las velocidades de los vehículos son reconstruidas según las comunas de origen y destino. Si para una ruta cualquiera existe uno o varios servicios que comparte las mismas comunas, entonces la velocidad se calcula como el promedio de las velocidades de dichos servicios. En el caso que no exista un servicio que reproduzca la ruta, se promedian los servicios que, en combinación, se parecen a la ruta.

Resulta interesante tener una idea de cómo se compara la estimación hecha con la estimación proporcionada por algún servicio de predicción de tráfico. Se decide usar para este fin las predicciones obtenidas por la API de Google Maps para una muestra pequeña de pares origen-destino. Como se observa en la figura 3.5, las estimaciones dadas por OSRM tienden a dar resultados parecidos con Google Maps, pero a medida que aumentan los tiempos de viaje, las diferencias se acentúan. Las estimaciones son tanto mayores como menores, sin embargo siguen una correlación con el servicio ofrecido por Google.

³<http://www.lasegunda.com/Noticias/Nacional/2012/11/796824/congestion-en-santiago-estudio-uc-u-de-chile-y-cepal-pronostica-que-velocidad-de-autos-bajara-20-en-2030>

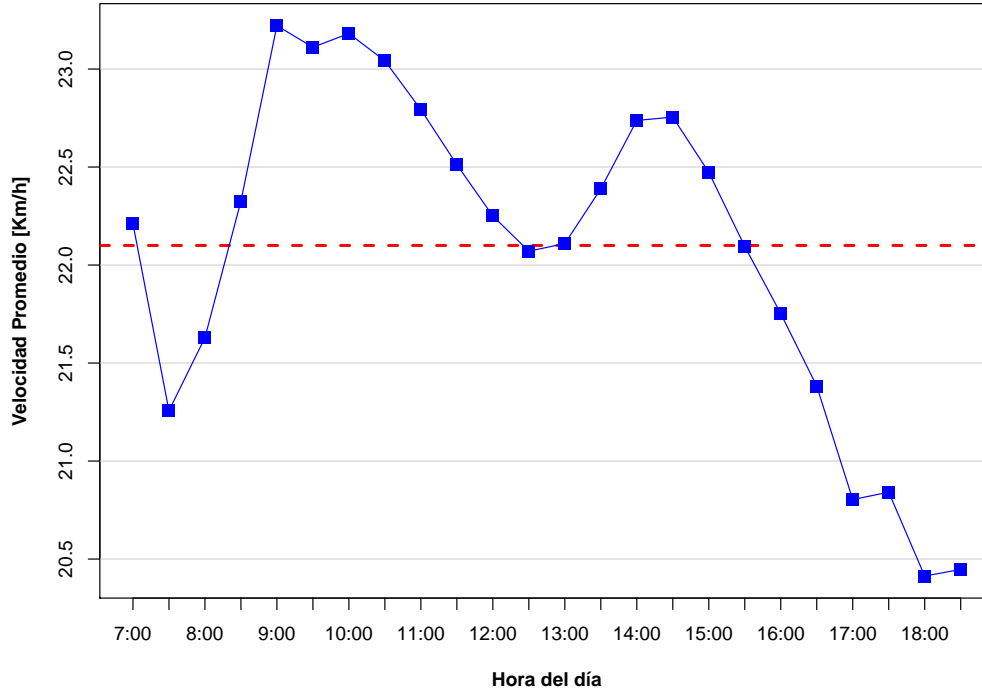


Figura 3.4: Velocidad promedio de los buses del Transantiago para el año 2010.

3.3. Modelo matemático

El modelo propuesto es adaptado del trabajo de Malandraki & Daskin sobre TDVRP[7]. También se introduce una variable propuesta por Cortes *et al.*[4] para postergar clientes en caso de ser conveniente.

Sea K el conjunto de técnicos disponibles. Los nodos a visitar están definidos por los conjuntos I_1, I_2 , con $I = I_1 \cup I_2$. Aquí I_1 representa los clientes que ya han sido postergados una vez y deben ser atendidos, I_2 representa los clientes no postergados sobre los que debe decidirse si son atendidos ahora o son postergados para el día siguiente. Los posibles nodos de partida y término de los recorridos se encuentran dentro de los conjuntos I_{ini} y I_{fin} , respectivamente. Por simplificación, I_{ini} y I_{fin} contienen un único nodo que es ficticio. Su propósito al introducirlos en el modelo es hacer funcionar las fórmulas de conservación de flujo. El conjunto M representa los intervalos de tiempo del día.

El tiempo de viaje del nodo i al nodo j queda expresado por el parámetro c_{ij}^m , donde el traslado comienza en el periodo m del día, con $c_{ij}^m = 0$, $i \in I_{ini}$, $j \in I_2 \cup I_{fin}$, $m \in M$. El tiempo de servicio de cada nodo es s_i , calculado como el tiempo promedio histórico de servicio para aquel tipo de solicitud, con $i \in I_1 \cup I_2$. La cota superior de cada intervalo de tiempo m está definido por T^m , con $T^0 = 0$, $m \in M$. La hora de inicio de la jornada laboral de los técnicos está dado por TI y la hora de término por TF .

Se define la variable de decisión binaria x_{ijk}^m igual a 1 si el técnico k inicia el viaje desde el

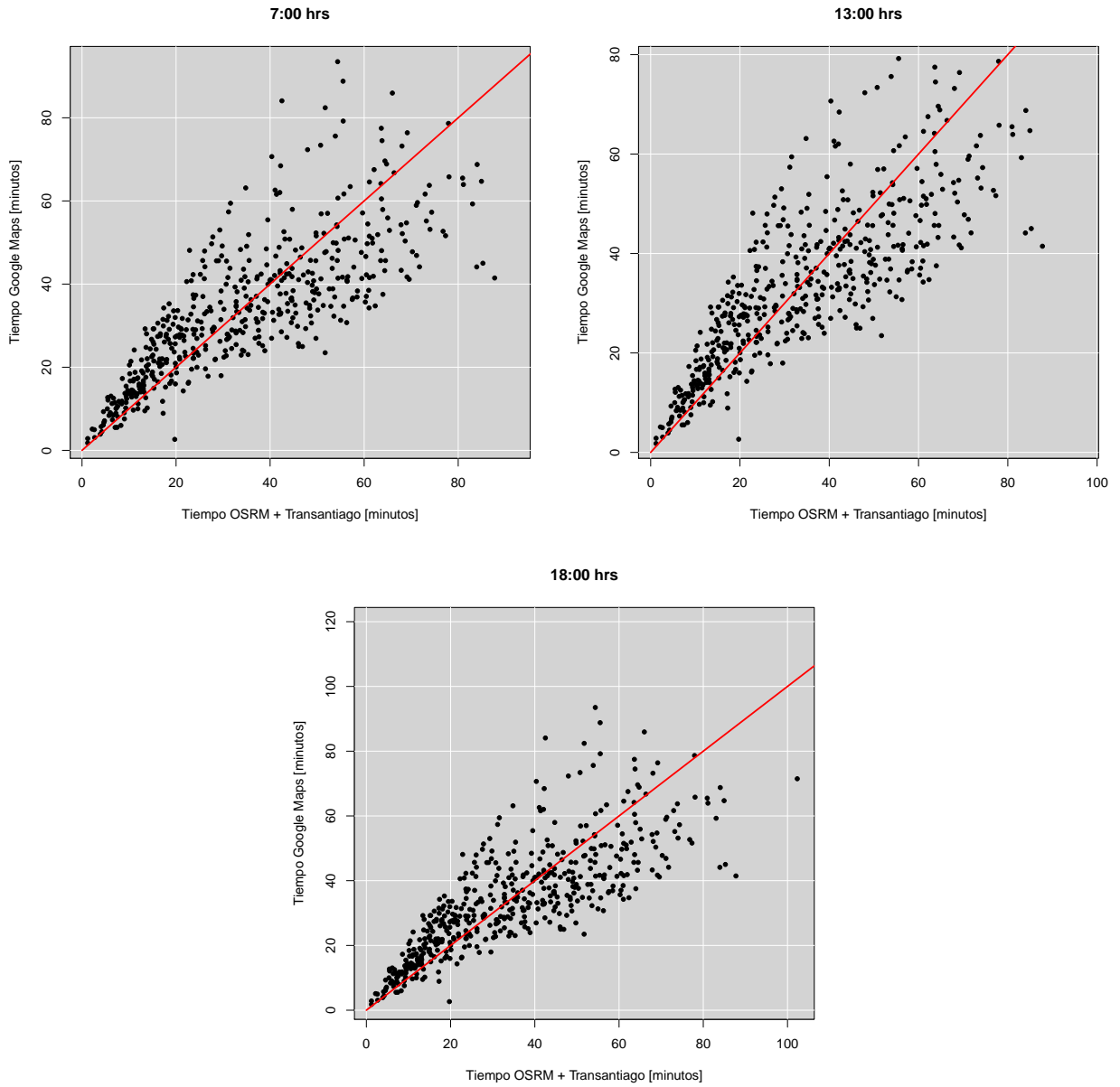


Figura 3.5: Comparación de 500 tiempos de viaje entre OSRM y Google Maps en distintas horas del día. Valores cuya diferencia de tiempo fueron mayores a 50 minutos fueron eliminados por ser *outliners* (2 valores).

nodo i hasta el nodo j en el periodo m y 0 si no, $i \in I_{ini} \cup I$, $j \in I \cup I_{fin}$, $k \in K$, $m \in M$. La variable entera t_i indica la hora en que cualquier técnico sale del nodo i , $i \in I$. Por último, se define la variable binaria v_i con valor 1 si el nodo i es postergado para el día siguiente y 0 si no, $i \in I$. El modelo matemático que representa el problema de despacho es:

$$\min_{x,t,v} \sum_{i \in I_{ini} \cup I} \sum_{j \in I \cup I_{fin}} \sum_{k \in K} \sum_{m \in M} c_{ij}^m x_{ijk}^m + \sum_{i \in I_2} P v_i \quad (3.1)$$

$$\text{s.a.} \quad \sum_{i \in I_{ini}} \sum_{j \in I \cup I_{fin}} \sum_{m \in M} x_{ijk}^m = 1 \quad k \in K \quad (3.2)$$

$$\sum_{i \in I_{ini} \cup I} \sum_{k \in K} \sum_{m \in M} x_{ijk}^m = 1 - v_j \quad j \in I \quad (3.3)$$

$$\sum_{i \in I_{ini} \cup I} \sum_{j \in I_{fin}} \sum_{m \in M} x_{ijk}^m = 1 \quad k \in K \quad (3.4)$$

$$\sum_{i \in I_1} v_i = 0 \quad (3.5)$$

$$t_i + c_{ij}^m + s_j - TF(1 - x_{ijk}^m) \leq t_j \quad \begin{array}{l} i \in I_{ini} \cup I, j \in I \cup I_{fin}, i \neq j, \\ k \in K, m \in M \end{array} \quad (3.6)$$

$$\sum_{i \in I_{ini} \cup I} \sum_{m \in M} x_{ijk}^m - \sum_{h \in I \cup I_{fin}} \sum_{m \in M} x_{jhm}^m = 0 \quad j \in I, k \in K \quad (3.7)$$

$$t_i - TF(1 - x_{ijk}^m) \leq T^m \quad \begin{array}{l} i \in I_{ini} \cup I, j \in I \cup I_{fin}, i \neq j, \\ k \in K, m \in M \end{array} \quad (3.8)$$

$$t_i - T^{m-1} x_{ijk}^m \geq 0 \quad \begin{array}{l} i \in I_{ini} \cup I, j \in I \cup I_{fin}, i \neq j, \\ k \in K, m \in M \end{array} \quad (3.9)$$

$$t_i \geq TI \quad i \in I_{ini} \quad (3.10)$$

$$t_i \leq TF \quad i \in I_{fin} \quad (3.11)$$

$$t_i \geq 0 \quad (3.12)$$

$$x_{ijk}^m, v_i \in \{0, 1\} \quad (3.13)$$

La función objetivo (3.1) busca reducir la suma del tiempo de viaje total de los técnicos y la penalización por cliente postergado. De ese modo un cliente puede ser programado para otros día si es costoso introducirlo en la ruta.

La restricción (3.2) asegura que los técnicos inician su ruta en el nodo ficticio de origen, mientras que la restricción (3.4) asegura que todas las rutas lleguen al nodo ficticio de término. La restricción (3.3) impone que cada cliente debe ser visitado una única vez por un solo técnico, siempre que ese cliente no sea postergado. Con la restricción (3.6) se asegura

la secuencia temporal del ruteo, evitando además el surgimiento de *subtours* dentro de la solución. La restricción (3.5) establece que los clientes postergados I_1 del día anterior deben ser atendidos en el mismo día. Debido a la existencia de varios periodos de viaje, puede ocurrir que el técnico que entra a un nodo sea distinto al que sale, siempre que los periodos de entrada y salida ocurran en diferentes horarios. Por este motivo se incluye la restricción 3.7, dando continuidad a los técnicos cuando visitan los nodos. Las restricciones (3.8) y (3.9) imponen que si el viaje comienza en el intervalo m , entonces los tiempos de salida t_i del nodo deben pertenecer también a dicho intervalo. Las horas de inicio y término de la jornada laboral están definidas por las restricciones (3.10) y (3.11), respectivamente. Finalmente, las restricciones (3.12) y (3.13) indican la naturaleza de las variables.

3.4. Valor de penalización

La función multiobjetivo (3.1) está compuesta de dos términos, donde por una parte se minimiza el tiempo de viaje de los técnicos, y por otra se minimiza la cantidad de clientes postergados en su atención. Aunque el tiempo de viaje es expresado en unidades de tiempo (minutos), el valor de penalización P no posee unidad de medida a primera vista. Lo único que resulta evidente es que su inclusión en el modelo obliga a disminuir las atenciones postergadas. A continuación se exploran ciertas interpretaciones sobre su significado dentro de la función objetivo.

3.4.1. Clasificación de clientes

No todos los clientes atendidos son iguales. Dentro de la cartera de clientes se pueden encontrar desde microempresas hasta grandes organismos como trasnacionales, instituciones públicas y universidades, entre otros. Para la empresa que ofrece el servicio técnico puede ser prioritario atender a sus clientes que, debido a su tamaño o modelo de negocios, requieren de una prestación rápida, mientras que los clientes domiciliarios pueden ser programados sin urgencia. De esta forma, el valor P representa la priorización que la empresa hace sobre sus clientes. Matemáticamente se expresa como

$$P_i = \text{Penalización por postergar cliente } i$$

, donde el valor se define de acuerdo a la segmento de clientes al cual pertenece i .

3.4.2. Tiempo de viaje esperado

Postergar un cliente implica que será atendido con obligatoriedad al día siguiente, es decir, deberá ser incluido en la programación de ruta de algún técnico, acortando el tiempo destinado a atender otros clientes del día. Específicamente, al tomar la decisión de postergar, se incurre en un costo de tiempo de viaje futuro. Este costo puede ser distinto del costo presente del viaje, tal como se muestra en la figura.

El tiempo diario que tiene cada técnico es limitado, así que la decisión de postergar se fundamenta en la expectativa del tiempo de viaje ahorrado. Para este caso, el valor P representa el tiempo de viaje futuro que empleará el técnico si atiende al cliente otro día.

Como las llamadas futuras no son conocidas, es imposible saber con exactitud cuál sería el costo de un cliente postergado. Una estimación para el valor P es el promedio de los tiempos de viaje pasados para el sector (comuna):

$$P_i = \sum_{i,j,m} c_{ij}^m x_{ij}^m / \sum_{i,j,m} x_{ij}^m.$$

La metodología expuesta considera que la función a minimizar es exclusivamente el tiempo de viaje, postergando los clientes bajo el criterio que ahorre tiempo y no porque sea el último recurso a utilizar.

3.4.3. Competitividad del mercado

Por lo general no existe una única empresa que ofrece un servicio determinado en el mercado, sino que existen varias empresas compitiendo por ofrece el mejor servicio. Dependiendo de la cantidad de competidores y clientes, el mercado podría ser muy competitivo, donde la empresa se vería obligada a ofrecer excelencia en la calidad de atención si quiere mantenerse en el negocio, o podría ser poco competitivo, donde podría relajar sus estándares de atención dada su posición dominante. En el caso particular de Xerox y el servicio técnico de equipos de oficina, la calidad de atención se expresa en la rapidez que ofrece para llegar donde los clientes. Si el mercado es muy competitivo, un retraso en la atención podría significar que ese cliente decida contratar los servicios de otros competidores. En cambio si el mercado es poco competitivo, la rapidez de atención dejaría de ser una prioridad para Xerox dado que existirían pocas alternativas para los clientes.

En este contexto el valor de penalización P representaría el nivel de competitividad con el que Xerox opera en el rubro. Un valor alto significaría que el mercado es competitivo, por lo que se privilegia la prontitud en la atención de clientes. Por otro lado, un valor bajo implica que el mercado es poco competitivo y se tiene mayor disposición a postergar clientes si es rentable.

3.4.4. Tasa de fuga

Comúnmente se asume que la calidad de servicio está relacionada con la retención de clientes. Una buena atención hace que los clientes vuelvan a solicitar el servicio, mientras que una mala atención no. A este último fenómeno se le llama “tasa de fuga”. Aplicando dicho principio al servicio técnico que ofrece Xerox, se puede relacionar la tasa de fuga con la postergación de su atención. Para este escenario, el valor P representa el costo esperado de perder un cliente si no es atendido durante el día. Matemáticamente se puede estimar como

$$P = \mathbb{E}[\text{valor del cliente}] \cdot \mathbb{P}[\text{fuga del cliente si no es atendido}]$$

La función objetivo se convierte en la suma de un término con unidades de distancia y otro con unidades monetarias. Por ese motivo, si se aplica la metodología expuesta, será conveniente darle un costo monetario a la distancia recorrida por los técnicos. Esto implica que el problema se convierte en la minimización de costos operativos.

Capítulo 4

Heurística

4.1. Idea general

La estrategia para resolver el problema de despacho de técnicos involucra dos técnicas: Generación de Columnas y Algoritmo Genético. Aunque el problema de Xerox ya ha sido resuelto de forma exacta en el trabajo de Cortés *et al.*[4], la cantidad de llamadas y técnicos disponibles que utilizaron para probar su modelo fueron mucho menores a las llamadas que pretenden planificarse en este trabajo, además que sus tiempos de resolución computacionales fueron mayores a una hora. Se modelaron instancias con 76 llamadas en total y 19 técnicos, mientras que el presente trabajo planificará 150 llamadas diarias aproximadamente. Por tal motivo se hace necesario una heurística para obtener soluciones.

El problema se plantea primero en términos de Generación de Columnas. Se define el PPL que representa el problema de despacho, llamado Problema Maestro, donde cada columna de la matriz de restricciones representa una ruta a elegir. Luego se define el PPL asociado, llamado Subproblema, que busca la ruta que más aporta a mejorar la solución. Resolver directamente el Subproblema es inviable por lo expuesto en el párrafo anterior, por lo tanto se resuelve mediante Algoritmo Genético, encontrando una ruta que mejore la función objetivo del Problema Maestro. Si tal ruta es encontrada, se agrega como nueva columna al Problema Maestro y la heurística vuelve a iterar. El ciclo inicia con el Problema Maestro dotado de unas pocas columnas que le permitan una solución inicial factible y termina cuando el Algoritmo Genético no encuentra rutas que mejoren la solución. Al finalizar se obtiene el PPL del despacho de técnicos, pero en una versión reducida tal que puede ser resuelta de forma exacta en muy poco tiempo. El esquema de la heurística para resolver el problema queda explicado en la figura 4.1.

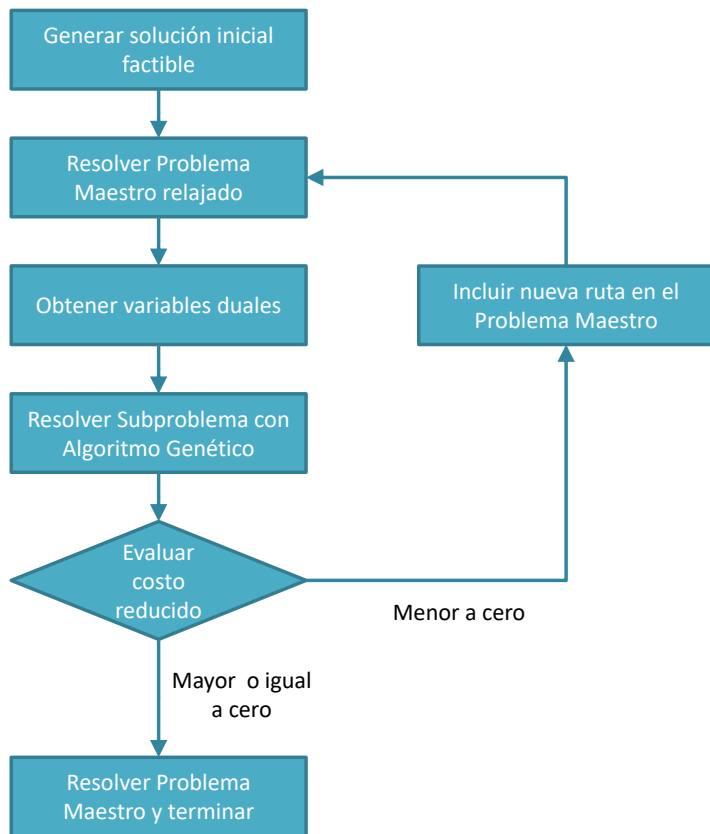


Figura 4.1: Esquema general de la heurística.

4.2. Generación de columnas

El PPL de despacho de técnicos se reescribe como un problema de *set partitioning* donde existe un conjunto fijo de rutas posibles a elegir para el ruteo. Sea R el conjunto de rutas posibles y θ_r variable binaria que vale 1 si se utiliza ruta r y 0 si no. Además sea v_i otra variable binaria igual a 1 si el nodo i (cliente) es postergado para el día siguiente y 0 de lo contrario. Sea a_{ir} una constante que vale 1 si ruta r pasa por el nodo i y 0 si no. Se define c_r como el costo de utilizar la ruta r en términos de tiempo de viaje. El Problema Maestro resultante es:

$$\min_{\theta, v} \sum_{r \in R} c_r \theta_r + \sum_{i \in I_2} P v_i \quad (4.1)$$

$$\text{s.a.} \sum_{r \in R} a_{ir} \theta_r = 1 \quad i \in I_1 \quad (4.2)$$

$$\sum_{r \in R} a_{ir} \theta_r + v_i = 1 \quad i \in I_2 \quad (4.3)$$

$$\sum_{r \in R} \theta_r = K \quad (4.4)$$

$$\theta_r, v_i \in \{0, 1\} \quad (4.5)$$

La función objetivo (4.1), como es usual, minimiza el costo total de la rutas y la penalización causada por postergar clientes. La restricción (4.2) asegura que cada cliente proveniente de la postergación del día anterior tiene prioridad de atención, mientras que la restricción (4.3) define si cada cliente del día es atendido o postergado. La restricción (4.4) impone que las rutas empleadas sean igual al número de técnicos disponibles. El Subproblema asociado para la columna a_{ir} , r fijo, busca minimizar el costo reducido \bar{c}_r de la ruta r . Si existe una ruta para la cual el costo reducido es menor a cero, entonces la introducción de aquella ruta al problema mejora la solución actual. El Suproblema se define como

$$\min_{c_r, a_{ir}} \bar{c}_r = c_r - \sum_{i \in I_1 \cup I_2} \alpha_i a_{ir} - \beta \quad (4.6)$$

$$\text{s.a. } r \text{ ruta factible} \quad (4.7)$$

donde α_i representa los valores duales de las restricciones (4.2) y (4.3), mientras que β es el valor dual de la restricción (4.4).

Los patrones factibles, en este caso rutas, serán generados mediante Algoritmo Genético. Con su implementación dentro del esquema de GC, esta heurística se preocupará solamente de generar rutas factibles, mientras que el Problema Maestro asegurará la combinación adecuada de ellas.

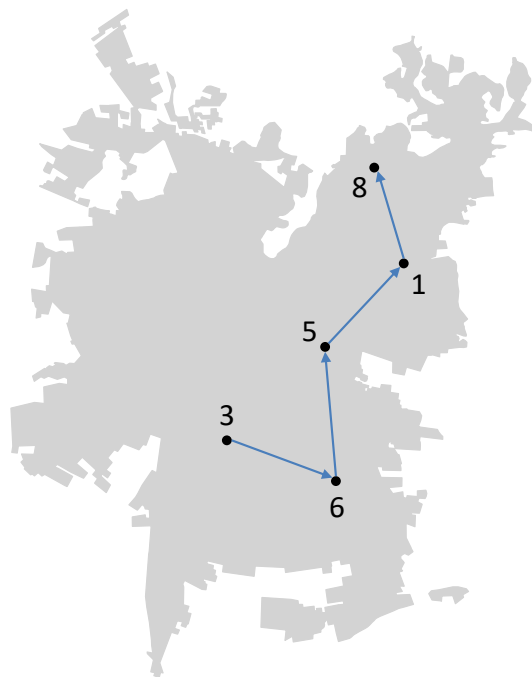
4.3. Algoritmo Genético

Los seres vivos en la naturaleza tienen como fin la supervivencia y reproducción. Las características físicas que le permiten su supervivencia en el medio ambiente son determinadas por su ADN, específicamente por unidades llamadas “genes”. Al reproducirse con otro organismo, los genes de ambos son traspasados a la descendencia, sin embargo esos genes no quedan intactos, sino que son recombinados aleatoriamente con el fin de crear una descendencia con características físicas distintas a las de sus progenitores. De todos los hijos solamente sobrevivirán los más aptos, es decir, los que posean la mejor combinación genética para sobrevivir en el medio. El modelo de Algoritmo Genético (AG) se basa en esta idea de reproducción biológica para encontrar soluciones a problemas de optimización. En términos generales, su esquema consiste en generar un conjunto de soluciones factibles del problema, seleccionar las mejores soluciones basado en alguna función de desempeño y luego recombinarlas varias veces hasta obtener el resultado deseado. Lo usual es que encuentre una solución aproximada del problema.

Una condición importante para aplicar AG a un problema es la capacidad de codificar las soluciones en “genes”, esto significa declararlas como una secuencia de números, similar a una secuencia genética. Para aplicarlo en el ruteo de vehículos, se define $L = (i_1, i_2, \dots, i_n)$ como la ruta a seguir por el técnico, donde i_j es el cliente i que es visitado en el orden j . En este caso, el gen queda definido como la posición j -ésima de la ruta L y la secuencia completa L representa el itinerario del técnico (ver figura 4.2). Notar que en este planteamiento, a diferencia del modelo matemático de la sección 3.3, se vuelve innecesario introducir nodos ficticios de partida y fin, puesto que la codificación no los requiere para construir rutas.

Se elige utilizar AG para resolver el Subproblema debido a lo simple de su programación y de los buenos resultados observados para resolver el VRP con dependencia del tiempo, con un *gap* máximo de 7% para instancias grandes[17]. A continuación se detallan las etapas del Algoritmo Genético para encontrar la solución aproximada del Subproblema, que corresponde a encontrar la ruta que minimice la función 4.6, mientras que su esquema se expone en la figura 4.4:

1. **Población inicial:** Se generan 4.000 rutas aleatorias compuestas de 8 clientes cada una, luego se convierten en rutas factibles cortando la secuencia donde la suma de los tiempos de viaje y servicio excede el tiempo máximo de trabajo. Formalmente, sea $L = (i_1, \dots, i_9)$ una ruta aleatoria y t_j el tiempo de salida del cliente en la j -ésima posición, este último calculado como la suma de los tiempos de viaje y servicio de la secuencia (i_1, \dots, i_j) , $j \in \{1, \dots, 10\}$. La ruta L es cortada en el primer cliente j cuya salida t_j es estrictamente mayor al horario de trabajo predefinido. El resultado es una sub-ruta factible de L .
2. **Función de ajuste:** La función que se busca minimizar es el costo reducido y por lo tanto será la función que evaluará la calidad de las rutas generadas. Su cálculo está dado por la fórmula 4.6. Como sus coeficientes dependen de las variables duales, esta fórmula cambiará sus valores en cada iteración que realice la Generación de Columnas.
3. **Selección:** De la población inicial se eligen los 400 caminos con el menor costo reducido para el Problema Maestro, luego se procede a realizar el *crossover*. El cálculo de los



$$P = (3, 6, 5, 1, 8)$$

Figura 4.2: Ejemplo de codificación de una ruta. Cada punto representa a un cliente i .

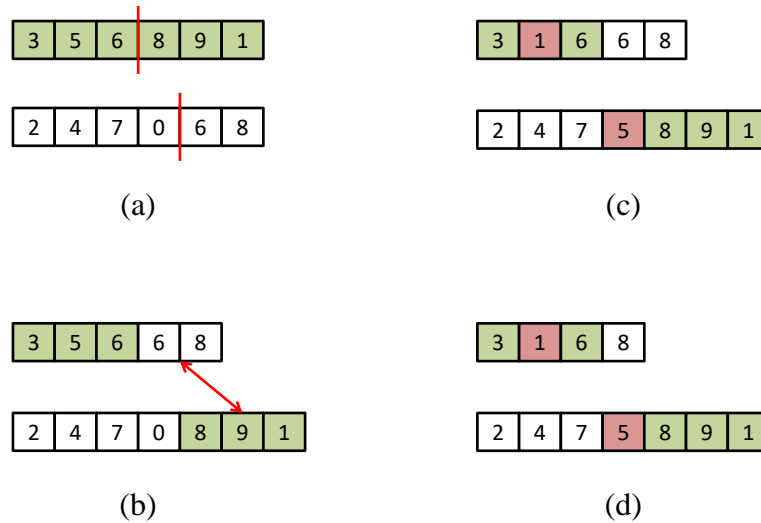


Figura 4.3: Ejemplo de operación *crossover* y mutación sobre dos rutas. (a) Se eligen dos rutas, donde la línea roja indica el punto de corte. (b) Se intercambian las secuencias. (c) Ciertos clientes sufren mutaciones. (d) Se eliminan los clientes repetidos en una misma ruta.

costos reducidos está dado por la fórmula 4.6.

4. **Crossover:** Hecha la selección del grupo más apto, se procede a reproducirlos entre sí 2.000 veces, creando en total 4.000 rutas. En cada operación de reproducción se eligen 2 rutas diferentes al azar, con 60 % de probabilidad de generar dos rutas con secuencias intercambiadas, tal como se muestra en la figura 4.3, o con 40 % de probabilidad de devolver las mismas rutas intactas. Formalmente, para una operación de *crossover*, sean $P_A = (i_1, \dots, i_n)$ y $P_B = (j_1, \dots, j_m)$ dos rutas factibles distintas elegidas de forma aleatoria. En ambas rutas se elige un punto de corte al azar $c_A \in \{1, \dots, n\}$ y $c_B = \{1, \dots, m\}$. Posteriormente se intercambian los segmentos, obteniendo como resultado un camino $P'_A = (i_1, \dots, i_{c_A}, j_{c_B+1}, j_m)$ y otro camino $P'_B = (j_1, \dots, j_{c_B}, i_{c_A+1}, i_n)$. Para finalizar, las rutas P'_A y P'_B son factibilizadas mediante el procedimiento expuesto en el primer paso.
5. **Mutación:** Cada cliente inserto en una ruta tiene un 1 % de probabilidad de ser reemplazado, esto es llamado “mutación”. Expresado de otra forma, dada una ruta P , la probabilidad de que el cliente i_j sea cambiado por otro i'_j es 1 %. Su ocurrencia es independiente de lo que suceda con otros clientes y con la probabilidad de *crossover*. Después de la mutación, el conjunto de rutas es nuevamente factibilizada.

Los resultados computacionales del AG mostraron ser lentos en su planteamiento original, por eso se introdujeron ciertas modificaciones para agilizar el cálculo. La primera es que la resolución del Subproblema, en vez de entregar la ruta con mejor costo reducido al Problema Maestro, entrega las 5 mejores, así aumenta la cantidad de columnas (rutas) construidas para el mismo tiempo de ejecución. Junto a esto se impone la condición de entregar la solución del

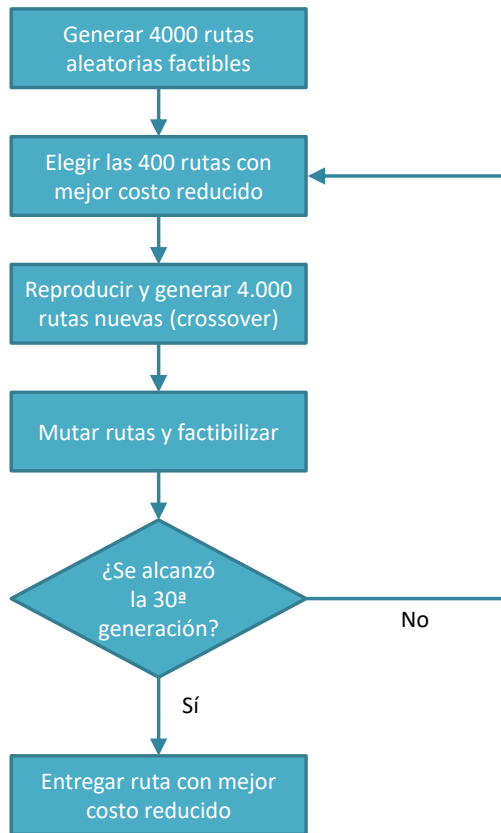


Figura 4.4: Esquema general del Algoritmo Genético propuesto.

Subproblema apenas la función objetivo sea negativa, sin esperar necesariamente a llegar a las 30 generaciones pero exigiendo un mínimo de 3 generaciones antes de evaluar. Por parte del Problema Maestro, el conjunto de soluciones se reevalúa cada 80 iteraciones. Cuando llega tal momento, se resuelve el problema entero y luego se eligen las 380 variables no-básicas con mayor costo reducido para ser eliminadas del modelo. Esta estrategia permite mantener acotado el conjunto de soluciones, facilitando la resolución del Problema Maestro en tiempo prudente.

Capítulo 5

Resultados

5.1. *Benchmark*

Es importante examinar el rendimiento de la heurística para justificar su aplicación. La forma de hacerlo es obtener la solución exacta para instancias muy pequeñas del problema y luego compararla con la obtenida por la heurística a probar. Experimentando con los datos, se descubrió que el problema podía ser resuelto de forma exacta y en tiempo prudente en instancias de hasta 13 clientes y 5 periodos de tiempo para 5 técnicos, por esto se impone como parámetro emplear solo 5 periodos de viaje. Las soluciones mostraron variaciones para distintos valores de penalización, sin embargo para valores mayores a 20 se mantenían idénticas, motivo por el cual se decidió usar dicho valor. También se descartó utilizar como medida de comparación la cota inferior del problema relajado dado que estaba considerablemente alejada de la solución entera. Debido a la dificultad de resolver el problema para una mayor cantidad de clientes, ya sea por tiempo de resolución o memoria de computador, se decide usar como punto de comparación los mejores resultados encontrados por CPLEX después de dos días de ejecución.

Clientes	Técnicos	Función Objetivo			Tiempo de Ejecución	
		MIP	AG	Gap	MIP	AG
10	2	112	119	5,9 %	460 seg.	41 seg.
15	3	138*	149	8,1 %	>2 días	245 seg.
20	4	127*	143	11,2 %	>2 días	214 seg.
25	5	129*	150	14,0 %	>2 días	577 seg.

*Mejor solución entera encontrada.

Cuadro 5.1: Comparación de resultados entre la solución exacta entera (**MIP**) y la solución obtenida por el algoritmo genético (**AG**). La muestra considera únicamente 5 periodos de viaje en la matriz de distancia.

Las pruebas se muestran en el cuadro 5.1. El *gap* entre el algoritmo y la solución exacta oscila en el rango 5-14 %, mientras que los tiempos de ejecución son notablemente inferiores en el caso de la heurística. Se puede apreciar que la solución exacta no pudo ser calculada para tres escenarios en menos de 2 días, en cambio el Algoritmo Genético encuentra soluciones menos de 10 minutos con cotas inferior para el *gap* entre 8-14 %. En conclusión se asume que los resultados significan un buen desempeño de la heurística para el poco tiempo de ejecución requerido.

5.2. Resultados computacionales

El Problema Maestro fue programado en el *software* AMPL[18] y resuelto mediante el *solver* IBM ILOG CPLEX 12.7[19], mientras que el Subproblema fue programado en el *software* R 3.4.0[20]. Los experimentos fueron ejecutados en una máquina con procesador Intel i5 2,4 GHz y 16 GB de memoria RAM.

Las instancias de prueba consisten en los primero 5 días hábiles de mayo del 2002. Este periodo destaca por su alta actividad, contabilizando un total de 803 llamadas de clientes y 56 técnicos despachados por día promedio (ver cuadro 5.2). La resolución del problema considera que los clientes postergados por el algoritmo son introducidos en el conjunto de clientes del día siguiente, con atención obligatoria. Esto significa que las decisiones tomadas por el algoritmo para un día afectan a las condiciones iniciales (clientes iniciales) de los días siguientes. Los valores de penalización se limitan hasta $P = 20$ pues, como fue mencionado en la sección anterior, no se observaron cambios en la solución para valores mayores.

# Técnicos	Clientes Atendidos	Clientes Postergados	Δ
56	331	472	59 %

Cuadro 5.2: Situación actual de la empresa para la primera semana de mayo.

Requisito importante para implementación del algoritmo es determinar en cuántos periodos se dividirá el día. Se define un periodo como el intervalo de tiempo dentro del día al cual le corresponde un tiempo de traslado diferente del resto de intervalos. Los datos proporcionados por Transantiago permiten dividir la jornada laboral hasta en 23 periodos de 30 minutos cada uno, con sus respectivos tiempos de viaje. En todas las instancias se probará dividir el día en periodos de media hora (23 intervalos) y una hora (12 intervalos), también en dividirlo en 2 periodos de alta congestión y otros 2 de baja congestión (4 intervalos), y por último considerar tiempos de traslado iguales durante el día, lo que es equivalente a usar un intervalo único.

Los cuadros 5.3–5.6 y la figura 5.1 muestran los resultados finales que se obtuvieron para las instancias considerando distintas penalizaciones y cantidades de técnicos disponibles, así como diferentes periodos para los tiempos de viajes. Como era previsto, la cantidad de clientes postergados disminuye a medida que aumenta la flota de técnicos. No obstante la disminución de las postergaciones es irregular, por ejemplo en el cuadro 5.5 para $P = 15$ se observa que

agregar 5 técnicos más disminuye en 27 los clientes postergados, pero si se vuelven a agregar otros 5 técnicos, solamente disminuyen en 9. Esto se explica porque la función objetivo busca disminuir simultáneamente clientes postergados y tiempo de viaje total, por lo que incluir más técnicos podría dar como resultado una planificación donde la cantidad de clientes atendidos no sea tanto como se espera, pero sí haya una reducción importante en el tiempo de viaje total.

Para valor de penalización $P = 10$, el más relajado, los retrasos en la atención de llamadas llegan a un rango de 11-20%. Esto quiere decir que apenas con 40 técnicos, 16 menos que la situación real, se logran reducir los retrasos más de la mitad si se compara que en la realidad Xerox operó con 56%. Para $P = 15$ el algoritmo alcanza las mejores planificaciones, logrando tasas de retrasos de 6-14%. Cuando la penalización es $P = 20$ se manifiesta un empeoramiento en las soluciones, llegando solamente a tasas de 8-12%. Una posible explicación a esto último puede estar en la forma que el valor P afecta al *solver*. Si es un valor muy grande, los cálculos de las variables duales que realice pierden precisión en sus decimales. Dado que el Algoritmo Genético depende de las variables duales para construir los coeficientes y evaluar las rutas, se tendría como consecuencia una elección ineficiente de las rutas que vuelve lenta las iteraciones.

Respecto del análisis de los tiempos de viaje variables, no se observó ningún patrón que sugiriera una mejora con respecto a un modelo con tiempos de viaje estáticos. La inclusión de dependencia del tiempo en el modelo, representados en los cuadros 5.3-5.5, no entregó soluciones diferentes a las entregadas por el modelo sin dependencia del tiempo, representado por el cuadro 5.6. Una posible explicación es que los tiempos de atención fueron mucho mayores que los tiempos de viaje. Los tiempos de atención mostrado en el cuadro 3.3 indican que gran parte de los tiempos superan los 40 minutos, llegando incluso a los 130 minutos, además el 48% de los clientes se concentran solamente en dos comunas como señala el cuadro 3.1. En comparación los traslado dentro del espacio geográfico reducido que representa cada comuna de Santiago no pasan de los 30 minutos de viaje. Ante tal magnitud de diferencia en ambos tiempo, resulta indiferente incluir tiempos de viaje variables al momento de planificar el ruteo.

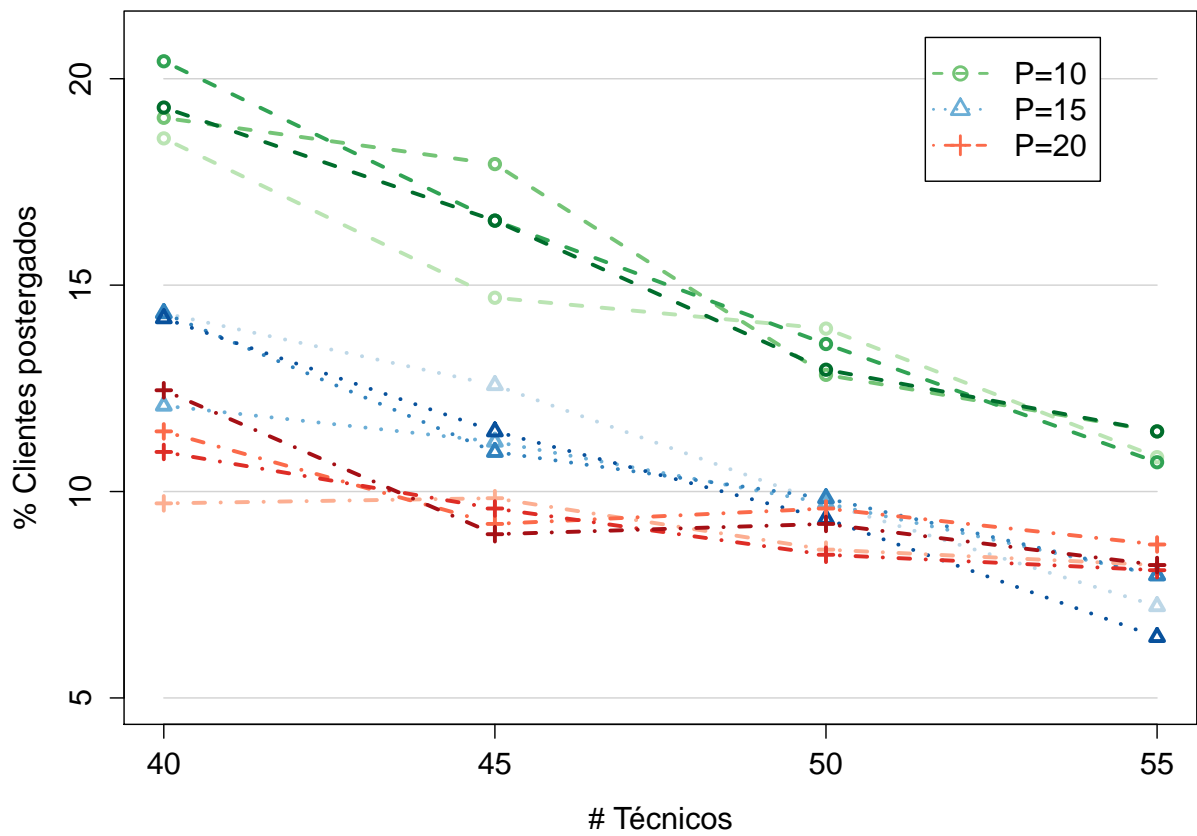


Figura 5.1: Resultados de las soluciones para clientes postergados. La tonalidad de las líneas indica la cantidad de periodos usados que dividen el día: tonos oscuros indican muchos periodos, tonos claros indican pocos periodos.

Resultados 23 Periodos						
#Tec	P	Función Objetivo	Clientes Atendidos	Clientes Postergados	Δ	Tiempo de Ejecución [seg]
40	10	3.090	745	149	19 %	6.834
45	10	2.644	748	118	15 %	6.708
50	10	2.459	758	112	14 %	6.801
55	10	2.039	764	87	11 %	6.728
40	15	3.752	732	115	14 %	7.050
45	15	3.356	740	101	13 %	6.983
50	15	2.921	761	78	10 %	6.828
55	15	2.533	772	58	7 %	6.739
40	20	3.933	744	78	10 %	7.155
45	20	3.877	741	79	10 %	6.995
50	20	3.568	753	69	9 %	7.010
55	20	3.322	752	66	8 %	7.077

#Tec = Cantidad de técnicos disponibles.

P = Valor de penalización.

Δ = Porcentaje de atenciones retrasadas.

Cuadro 5.3: Resultados obtenidos para el despacho de técnicos con distintas cantidades de técnicos y valores de penalización para velocidades cada 30 minutos (23 periodos en la matriz de viaje).

Resultados 12 Periodos						
#Tec	P	Función Objetivo	Clientes Atendidos	Clientes Postergados	Δ	Tiempo de Ejecución [seg]
40	10	3.127	738	153	19 %	8.403
45	10	2.805	742	144	18 %	8.326
50	10	2.385	759	103	13 %	8.256
55	10	2.132	771	92	11 %	8.234
40	15	3.627	745	97	12 %	8.484
45	15	3.379	750	90	11 %	8.324
50	15	2.898	753	78	10 %	8.309
55	15	2.612	771	64	8 %	8.247
40	20	4.289	738	92	11 %	8.846
45	20	4.105	748	74	9 %	8.438
50	20	3.723	744	77	10 %	8.479
55	20	3.565	753	70	9 %	8.674

#Tec = Cantidad de técnicos disponibles.

P = Valor de penalización.

Δ = Porcentaje de atenciones retrasadas.

Cuadro 5.4: Resultados obtenidos para el despacho de técnicos con distintas cantidades de técnicos y valores de penalización para velocidades cada una hora (12 periodos en la matriz de viaje).

Resultados 4 Periodos						
#Tec	P	Función Objetivo	Clientes Atendidos	Clientes Postergados	Δ	Tiempo de Ejecución [seg]
40	10	3.197	732	164	20 %	6.855
45	10	2.790	752	133	17 %	6.754
50	10	2.358	755	109	14 %	6.641
55	10	2.123	770	86	11 %	6.616
40	15	3.642	734	115	14 %	7.262
45	15	3.076	746	88	11 %	6.692
50	15	2.810	753	79	10 %	6.705
55	15	2.620	768	64	8 %	6.722
40	20	4.150	733	88	11 %	7.347
45	20	3.853	745	77	10 %	7.043
50	20	3.556	753	68	8 %	6.976
55	20	3.557	760	65	8 %	6.940

#Tec = Cantidad de técnicos disponibles.

P = Valor de penalización.

Δ = Porcentaje de atenciones retrasadas.

Cuadro 5.5: Resultados obtenidos para el despacho de técnicos con distintas cantidades de técnicos y valores de penalización para 4 periodos en la matriz de viaje.

Resultados Periodo Único						
#Tec	P	Función Objetivo	Clientes Atendidos	Clientes Postergados	Δ	Tiempo de Ejecución [seg]
40	10	3.087	749	154	19 %	6.642
45	10	2.793	751	133	17 %	6.534
50	10	2.383	758	104	13 %	6.436
55	10	2.078	764	92	11 %	6.397
40	15	3.754	736	114	14 %	6.854
45	15	3.371	751	92	11 %	6.587
50	15	2.902	762	75	9 %	6.441
55	15	2.582	781	52	6 %	6.562
40	20	4.321	725	100	12 %	6.947
45	20	3.796	753	72	9 %	6.806
50	20	3.785	749	74	9 %	6.653
55	20	3.551	755	66	8 %	6.733

#Tec = Cantidad de técnicos disponibles.

P = Valor de penalización.

Δ = Porcentaje de atenciones retrasadas.

Cuadro 5.6: Resultados obtenidos para el despacho de técnicos con distintas cantidades de técnicos y valores de penalización para un único periodo en la matriz de viaje, correspondiente a la velocidad promedio del día.

Capítulo 6

Conclusión

Resolver el problema lineal del VRP puede resultar excesivamente costoso en tiempo, incluso para instancias relativamente pequeñas (13 clientes). Más aún, cuando se incluye dependencia del tiempo a la matriz de viaje, el problema resulta imposible de resolver de forma exacta. Agregarle más periodos de tiempo aumenta la complejidad del problema, haciendo necesario buscar otras estrategias para encontrar una solución matemática aceptable.

El Algoritmo Genético, en conjunto con la Generación de Columnas, demostró ser un procedimiento rápido y de buenos resultados aproximados cuando se quiere resolver el TDVRP. Haciendo una comparación con la situación de Xerox para una semana ocupada, el algoritmo encuentra soluciones que reducen la cantidad de clientes postegados de un 59 % actual a un 6-20 %. Sin embargo, debe considerarse que la solución teórica no abarca posibles inconvenientes que ocurren en la realidad: técnicos que presentan averías en su vehículo, tiempos reales de servicio mucho mayores a los esperados, ausencia de algunos técnicos, etc. Lo que debe interpretarse es que una buena planificación de la flota puede mejorar la calidad de atención en general.

La incorporación de tiempos de viaje dependientes de la hora del día no tuvo ningún impacto en las soluciones finales respecto de soluciones convencionales. La causa sería que los tiempos de servicio son notablemente mayores que los tiempos de viaje, haciendo que el impacto de estos últimos sean irrelevantes para la solución del modelo en una ciudad. Posiblemente el problema de despacho de técnicos se vería beneficiado con dependencia del tiempo si los tiempos de servicio fueran menores a los tiempos de viaje.

Bibliografía

- [1] Cristián E Cortés, Michel Gendreau, Daniel Leng, and Andres Weintraub. A simulation-based approach for fleet design in a technician dispatch problem with stochastic demand. *Journal of the Operational Research Society*, 62(8):1510–1523, 2011.
- [2] José Antonio Rojas Mejias. Heurística basada en covering y generacion de columnas dinamico para el problema de ruteo dinámico de vehículos con ventanas de tiempo. Tesis de Magíster, Universidad de Chile, Santiago de Chile, 2012.
- [3] Sebastián Souyris, Cristián E Cortés, Fernando Ordóñez, and Andres Weintraub. A robust optimization approach to dispatching technicians under stochastic service times. *Optimization Letters*, 7(7):1549–1568, 2013.
- [4] Cristián E Cortés, Michel Gendreau, Louis Martin Rousseau, Sebastián Souyris, and Andrés Weintraub. Branch-and-price and constraint programming for solving a real-life technician dispatching problem. *European Journal of Operational Research*, 238(1):300–312, 2014.
- [5] Paulina Andrea Briceño Aguirre. Heurística basada en predicción de demanda y generación de columnas para resolver el problema de ruteo dinámico de vehículos con ventanas de tiempo. Tesis de Magíster, Universidad de Chile, Santiago de Chile, 2014.
- [6] Michel Gendreau, Gianpaolo Ghiani, and Emanuela Guerriero. Time-dependent routing problems: A review. *Computers & Operations Research*, 64:189–197, 2015.
- [7] Chryssi Malandraki and Mark S. Daskin. Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transportation science*, 26(3):185–200, 1992.
- [8] Chryssi Malandraki. *Time dependent vehicle routing problems: Formulations, solution algorithms and computational experiments*. Tesis de Doctorado, Northwestern University, Illinois, Estados Unidos, 1989.
- [9] Soumia Ichoua, Michel Gendreau, and Jean-Yves Potvin. Vehicle dispatching with time-dependent travel times. *European journal of operational research*, 144(2):379–396, 2003.
- [10] Suresh Nanda Kumar and Ramasamy Panneerselvam. A time-dependent vehicle routing problem with time windows for e-commerce supplier site pickups using genetic algorithm.

Intelligent Information Management, 7(04):181, 2015.

- [11] Éric Taillard, Philippe Badeau, Michel Gendreau, François Guertin, and Jean-Yves Potvin. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science*, 31(2):170–186, 1997.
- [12] Arthur V. Hill and W.C. Benton. Modelling intra-city time-dependent travel speeds for vehicle scheduling problems. *Journal of the Operational Research Society*, 43(4):343–351, 1992.
- [13] Marco E. Lübbecke and Jacques Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.
- [14] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.
- [15] Mordechai Haklay. How good is volunteered geographical information? a comparative study of openstreetmap and ordnance survey datasets. *Environment and planning B: Planning and design*, 37(4):682–703, 2010.
- [16] Dennis Luxen and Christian Vetter. Real-time routing with openstreetmap data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pages 513–516, New York, NY, USA, 2011. ACM.
- [17] Soojung Jung and Ali Haghani. Genetic algorithm for the time-dependent vehicle routing problem. *Transportation Research Record: Journal of the Transportation Research Board*, 1771:164–171, 2001.
- [18] Robert Fourer, David M Gay, and Brian W Kernighan. A modeling language for mathematical programming. *Management Science*, 36(5):519–554, 1990.
- [19] IBM ILOG CPLEX. *CPLEX User's Manual. Version 12.7*. IBM Corp., 2017.
- [20] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.