

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE POSGRADO EN INGENIERÍA Y CIENCIAS

**OPTIMIZACIÓN DE RECORRIDOS PARA VISITAS TÉCNICAS DE
UNA EMPRESA DE SERVICIOS**

**PROYECTO PREVIO A LA OBTENCIÓN DEL GRADO DE MAGÍSTER
EN INVESTIGACIÓN OPERATIVA**

JEYSSON VINUEZA MUÑOZ

DIRECTOR: DR. LUIS MIGUEL TORRES

MARZO, 2006

DECLARACIÓN

Yo, Jeysson Vasili Vinueza Muñoz, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

JEYSSON VINUEZA MUÑOZ

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Jeysson Vinueza Muñoz, bajo mi supervisión.

Dr. Luis Miguel Torres
DIRECTOR DE PROYECTO

AGRADECIMIENTOS

Agradezco de manera especial al director de proyecto quien con su acertada guía me permitió llevar a feliz término este estudio.

DEDICATORIA

A Miriam, Erik y Alex.

RESUMEN

El presente trabajo es una aplicación de la teoría de enrutamiento de vehículos (VRP) para mejorar los recorridos que el personal de una empresa recorre día a día para brindar servicio técnico.

Esta empresa se dedica al monitoreo de sistemas de seguridad y una de sus principales metas es brindar un servicio con altos estándares de calidad. Debido a la naturaleza del negocio la respuesta por parte del personal técnico debe ser inmediata.

En la actualidad, debido al creciente número de clientes, el tiempo de respuesta ante contingentes en los sistemas de seguridad ha aumentado más allá de sus estándares. Esto ha motivado a sus directivos a buscar alternativas de mejoramiento.

Ante esta realidad, las herramientas matemáticas para el enrutamiento juegan un papel muy importante pues puede ayudar a la empresa a optimizar el tiempo que sus técnicos dedican a brindar el servicio, mediante la minimización de los tiempos de desplazamiento entre clientes. Esto evidentemente le permitirá a la empresa conseguir mejores resultados en sus operaciones y procesos de soporte técnico.

Los resultados obtenidos en los ejemplos de prueba, si bien no redujeron el número de técnicos, permitieron que al menos uno de ellos disponga de tiempo para hacer trabajos de oficina y dar servicio técnico emergente a problemas que se susciten durante el día. Además de balancear el trabajo entre los técnicos se logró atender, en la mayoría de casos, a todos los clientes que requerían servicio ese día.

PRESENTACIÓN

El presente trabajo se desarrolló partiendo del contacto inicial con la empresa, la misma que manifestó su problemática actual con respecto a la velocidad de respuesta ante problemas con los sistemas de seguridad en sus clientes. Se revisaron varias opciones de mejoramiento y se concluyó que lo mejor sería atacar el tiempo de desplazamiento de sus técnicos hacia los clientes.

Una vez definida la temática se revisó la bibliografía concerniente al enrutamiento de vehículos y formas de abordar este tipo de problemas. Se revisaron varios modelos de programación matemática y se eligió uno de ellos. Para resolver estos modelos fue necesario buscar en el Internet software aplicable a este tipo de modelos.

Para implementar estos modelos fue necesario construir la red digital de la ciudad de Quito, al menos en la zona de interés, para poder modelar los desplazamientos. Esta tarea tomó más tiempo del previsto pues fue necesario recorrer la ciudad para verificar el sentido de las calles, barreras físicas y señalizaciones de tránsito. Adicionalmente, fue necesario ubicar geográficamente en el mapa digital de la ciudad a los clientes y a la oficina principal.

Una vez implementado el modelo se hicieron pruebas sobre ejemplos concretos y se presentaron los resultados de los mismos. En la parte final se elaboraron las conclusiones del estudio y sugerencias para trabajos futuros en esta área.

Contenidos

1	Introducción	1
1.1	Datos de la empresa	2
1.2	Alcance del presente estudio	2
1.3	Situación Actual	3
1.3.1	Proceso para obtener servicio técnico	3
1.3.2	Parámetros actuales y de servicio	3
1.3.3	Problemas detectados	4
1.4	Requerimientos del modelo de optimización	4
2	Definiciones y notaciones	6
3	Problema de Enrutamiento de Vehículos (VRP)	11
3.1	Introducción	11
3.2	Modelos para solucionar el VRP	12
3.2.1	Modelo basado en arcos	12
3.2.2	Modelo basado en rutas	14
3.3	Variantes del VRP	15
3.3.1	VRP con ventanas de tiempo	15
3.3.2	VRP con recargas	16
3.3.3	VRP con cargas y descargas	16
3.4	Heurísticas clásicas para el VRP	17
3.4.1	Introducción	17
3.4.2	Métodos Constructivos	18
3.4.3	Heurísticas de inserción secuencial	19
3.4.4	Métodos de dos fases	21
3.4.5	Heurísticas de mejoramiento	22

4	Modelo de P.E. para el problema de rutas	25
4.1	Conjuntos necesarios	25
4.2	Red auxiliar	25
4.2.1	Definición	25
4.2.2	Utilidad	26
4.3	Red principal	26
4.3.1	Definición	26
4.4	Consideraciones de tiempo	27
4.5	Compatibilidad	28
4.6	Variables de decisión	28
4.7	Modelo de optimización preliminar MOP	29
4.8	Modelo de programación entera MPE	30
4.9	Modelo con ventanas de tiempo suaves MVTS	30
4.10	Heurística de Inserción	31
5	Implementación computacional	33
5.1	Localización geográfica de vértices y arcos	33
5.1.1	Localización de los clientes	33
5.1.2	Localización de la oficina central	33
5.1.3	Puntos de tránsito	34
5.1.4	Arcos	35
5.2	Características del hardware y software	36
5.2.1	Hardware	36
5.2.2	Software	36
5.3	Base de datos de la red de Quito y clientes	37
5.3.1	Red de Quito	37
5.3.2	Localización de clientes	38
5.3.3	Estructura de la base de datos	40
5.3.4	Diseño de la base de datos	40
5.4	Construcción de la red auxiliar	40
5.5	Construcción de la red principal	45
5.6	Implementación de los modelos de programación entera	46
5.6.1	MPE	46
5.6.2	MVTS	46

6	Resultados computacionales	48
6.1	Origen de los datos	48
6.2	Ejemplos de prueba pequeños (15 clientes y 2 técnicos)	48
6.2.1	Comparación de los resultados computacionales de los modelos MPE y MVTs	48
6.2.2	Comparación de los itinerarios resultantes de los modelos MPE y MVTs	49
6.2.3	Rutas	55
6.3	Ejemplos de prueba con 60 clientes y 5-7 técnicos	56
6.3.1	Resultados computacionales del modelos MVTs	56
6.3.2	Itinerario resultante del modelo MVTs	56
6.3.3	Rutas	56
7	Conclusiones y trabajo futuro	60
7.1	Conclusiones	60
7.2	Trabajo futuro	61
A	Datos	62
A.1	Tabla de vértices	62
A.2	Tabla de arcos	63
A.3	Tabla de Vértices - Cliente	64
B	Puntos de tránsito y movimientos factibles	65
B.1	Puntos de tránsito	65
B.1.1	Cruces sin restricciones	65
B.1.2	Cruces con restricciones	66
B.2	Movimientos factibles	67
B.2.1	Sentido de las calles	67
B.2.2	Barreras arquitectónicas y accidentes geográficos	67
B.2.3	Señales prohibitivas de circulación	68
C	Modelo GLPK	71
C.1	Modelo MPE	71
C.1.1	Modelo en GLPK	71
C.1.2	Solución del modelo en GLPK	74
C.2	Modelo MVTs	75
C.2.1	Modelo en GLPK	75

D Mapas Resultantes	78
D.1 Rutas por cada técnico (15 clientes)	78
D.2 Rutas por cada técnico (60 clientes)	78

Listado de Figuras

2.1	Ejemplo de representación gráfica de una red	7
2.2	Ejemplo de una subred	8
2.3	Ejemplo de una subred inducida	8
2.4	Ejemplo de caminos y circuitos en una red	9
3.1	Cruce de secuencias	23
3.2	Intercambio de secuencias	23
3.3	Relocalización de secuencias	23
4.1	Ejemplo del costo de un arco en la red R	27
5.1	Localización de los clientes en el mapa de Quito	34
5.2	Ejemplo de puntos de tránsito	35
5.3	Vértices y arcos sobre el mapa digital	37
5.4	Clientes sobre el mapa digital	39
5.5	Diagrama entidad relación de la base de datos	41
6.1	Gráfico comparativo entre los modelos MPE y MVTs en tiempos totales (15 clientes)	51
6.2	Gráfico comparativo entre los modelos en tiempos sobrantes (15 clientes)	52
6.3	Itinerario del EJM-15-01 en el modelo MPE	53
6.4	Itinerario del EJM-15-02 en el modelo MPE	53
6.5	Itinerario del EJM-15-01 en el modelo MVTs	54
6.6	Itinerario del EJM-15-02 en el modelo MVTs	55
6.7	Gráfico de técnicos necesarios en cada caso y sus tiempos sobrantes .	58
6.8	Gráfico de técnicos necesarios en cada caso y sus tiempos sobrantes .	59
B.1	Ejemplo de movimientos entre puntos de tránsito - sentido de calles . .	67

B.2	Ejemplo de movimientos entre puntos de tránsito - barreras arquitectónicas	68
B.3	Ejemplo de movimientos factibles en un cruce - situación real	69
B.4	Ejemplo de movimientos factibles en un cruce - movimientos factibles N-S	69
B.5	Ejemplo de movimientos factibles en un cruce	69
B.6	Ejemplo de movimientos factibles en un cruce complejo (10 de Agosto y Eloy Alfaro)	70
D.1	Mapa de las rutas del ejemplo EJM-15-01 (MPE)	78
D.2	Mapa de las rutas del ejemplo EJM-15-01 (MVTS)	79
D.3	Mapa de las rutas del ejemplo EJM-15-02 (MPE)	79
D.4	Mapa de las rutas del ejemplo EJM-15-02 (MVTS)	80
D.5	Mapa de las rutas del ejemplo EJM-15-03 (MPE)	80
D.6	Mapa de las rutas del ejemplo EJM-15-03 (MVTS)	81
D.7	Mapa de las rutas del ejemplo EJM-15-04 (MPE)	81
D.8	Mapa de las rutas del ejemplo EJM-15-04 (MVTS)	82
D.9	Mapa de las rutas del ejemplo EJM-60-01 (MVTS)	83
D.10	Mapa de las rutas del ejemplo EJM-60-02 (MVTS)	84
D.11	Mapa de las rutas del ejemplo EJM-60-03 (MVTS)	85
D.12	Mapa de las rutas del ejemplo EJM-60-04 (MVTS)	86

Listado de Tablas

5.1	Tablas de la base de datos	41
5.2	Lista de campos de la tabla R01_Vertices	42
5.3	Lista de campos de la tabla R02_Arcos	42
5.4	Lista de campos de la tabla R03_VerticesCliente	42
5.5	Lista de campos de la tabla R04_ArcosGCompleto	42
5.6	Lista de campos de la tabla R05_CasosEjemplo	43
5.7	Lista de campos de la tabla R06_ClientesEjemplo	44
5.8	Lista de campos de la tabla R07_SolucionXij	44
6.1	Comparación de los modelos MPE y MVTS para ejemplos pequeños	50
6.2	Itinerario del EJM-15-01 en el modelo MPE	51
6.3	Itinerario del EJM-15-02 en el modelo MPE	52
6.4	Itinerario del EJM-15-01 en el modelo MVTS	54
6.5	Itinerario del EJM-15-02 en el modelo MVTS	55
6.6	Resultados computacionales del modelo MVTS sobre 60 clientes	57
A.1	Resumen de la tabla R01_Vertices	62
A.2	Resumen de la tabla R02_Arcos	63
A.3	Resumen de la tabla R03_VerticesCliente	64

Capítulo 1

Introducción

La globalización ha transformado la forma en que las empresas llevan a cabo sus operaciones, pero sobre todo, ha tenido un impacto muy importante en la organización misma, por lo que el perfil de sus integrantes es distinto. La notable reducción de los costos del transporte y las comunicaciones ha facilitado la división del proceso productivo, permitiendo la participación de un mayor número de localizaciones geográficas según las ventajas que cada una aporta a la cadena de valor agregado. Este hecho ha ampliado las oportunidades para que economías individuales participen más activamente de las redes internacionales de producción administradas por las grandes compañías multinacionales. El proceso de globalización también crea nuevas oportunidades en tanto incrementa la competencia, sienta las bases para el establecimiento de nuevas alianzas empresarias y sociales, y contribuye a la desarticulación de los oligopolios establecidos.

Ante esta realidad, las herramientas matemáticas juegan un papel muy importante pues pueden ayudar a las empresas a conseguir mejores resultados en sus operaciones y procesos. Alrededor del mundo nuevas tecnologías están en constante desarrollo brindando a la industria nuevas posibilidades de mejorar sus negocios y ser más competitivas.

Laarcom Cía. Ltda. es una empresa dedicada al monitoreo y supervisión de sistemas de seguridad. Debido al constante crecimiento de la inseguridad en las ciudades y al deseo de las personas y empresas de protegerse han aparecido otras empresas dedicadas a proveer de este servicio.

La competencia es cada vez más dura por lo que las empresas se han visto en la necesidad de mejorar sus índices de calidad. Uno de los principales índices mide el tiempo de respuesta de la empresa ante las siguientes situaciones:

- un daño en el sistema de seguridad del cliente
- la necesidad de ampliar un sistema de seguridad en un cliente
- el traslado de un sistema de seguridad de un lugar a otro

La empresa maneja este índice con valores de 24 a 48 horas. últimamente el tiempo de respuesta ante las necesidades de los clientes ha superado estos valores llegando en muchos casos hasta los cuatro días.

Esto ocasiona, a más de las justificadas quejas de los clientes, una mala imagen para la empresa lo que va en contra de su razón de ser que es la de brindar un servicio con altos niveles de seguridad, confianza y calidad a sus clientes.

El presente estudio busca mejorar este índice de calidad de servicio. Una mejora en este aspecto se logrará enviando un técnico, lo antes posible, al lugar donde se requiere su servicio. Es aquí donde la Matemática juega un papel importante en la optimización, pues se buscará un modelo que minimice los tiempos que los técnicos necesitan para trasladarse de un cliente a otro.

Como consecuencia de esto se atenderá a más clientes con tiempos de espera menores y los técnicos podrán utilizar sus tiempos sobrantes en tareas de oficina.

1.1 Datos de la empresa

Laarcom Cía. Ltda. está domiciliada en la ciudad de Quito, su principal actividad es el monitoreo de alarmas vía telefónica y por radio. La mayor parte de sus clientes se encuentran distribuidos en las ciudades de Quito y Guayaquil. Existen clientes en otras ciudades del Ecuador, pero en menor número.

En el caso de la ciudad de Quito la mayoría de clientes se localizan en una franja comprendida entre la avenida Patria y la avenida Naciones Unidas. Un porcentaje menor de sus clientes se encuentra distribuido en el sector norte y sur de Quito, en los valles de Cumbayá-Tumbaco y en el valle de los Chillos.

1.2 Alcance del presente estudio

El presente estudio se limitará a los clientes que la empresa mantiene en la parte central de la ciudad de Quito. La problemática a abordar se centrará en mejorar el tiempo de respuesta del equipo técnico de la empresa para los requerimientos de los clientes mediante de la minimización de los tiempos de traslado.

1.3 Situación Actual

1.3.1 Proceso para obtener servicio técnico

Existen dos maneras en que un cliente obtiene servicio técnico:

1. Cuando el personal de monitoreo detecta alguna falla en el sistema de seguridad contacta al cliente y le notifica el problema. Como resultado de esta gestión el cliente acepta la visita de un técnico y la novedad es ingresada en el sistema.
2. En el momento que un cliente tiene un problema con su sistema de seguridad (sensores y cámaras dañados, cables en mal estado, alarma que no responde, etc.) se contacta con la empresa y solicita la visita de un técnico. La empresa verifica que el cliente se encuentre al día con sus obligaciones económicas y registra el tipo de requerimiento en su sistema informático. Al cliente se le solicitan los siguientes datos:
 - Tipo de problema
 - Datos de la persona que llama

En cualquiera de los dos casos anteriores se sigue el siguiente procedimiento:

1. Se registran durante el día todas las novedades en el sistema informático.
2. A primera hora del día siguiente el encargado del departamento técnico genera la lista de clientes a ser visitados.
3. De manera manual el encargado distribuye el trabajo entre los técnicos y genera una hoja de ruta para cada uno.
4. Los técnicos visitan a los clientes y entregan los reportes de visitas al final del día. Los clientes que fueron visitados se retiran de la lista de pendientes del sistema informático. Aquellos clientes que no fueron visitados se mantienen pendientes para ser visitados al día siguiente.

1.3.2 Parámetros actuales y de servicio

1. Los clientes son visitados al día siguiente de haber reportado el problema.
2. Los técnicos tienen prefijadas zonas de la ciudad de Quito para brindar el servicio.

3. El número de técnicos a ser asignados puede cambiar de un día a otro, dependiendo del número de solicitudes de servicio que se tengan.
4. Aproximadamente el 10% de los clientes solicitan la visita de un técnico en un horario prefijado.
5. Un promedio de 60 clientes solicitan servicio cada día.

1.3.3 Problemas detectados

Los principales problemas que afectan a la empresa al momento de asignar los técnicos para hacer las visitas a los clientes son los siguientes:

1. Los clientes que requieren servicio varían día a día. Esto implica que los técnicos deben desplazarse por zonas diferentes cada día de acuerdo a los lugares de domicilio de los clientes.
2. Cada día algunas de las zonas de la ciudad tienen más clientes que otras por lo que la carga de trabajo entre los técnicos que atienden a esas zonas no es uniforme.
3. Como consecuencia de las distancias que deben desplazarse los técnicos, algunos técnicos no visitan a todos los clientes de su hoja de ruta, mientras que otros terminan su trabajo antes de tiempo.
4. Algunos clientes son visitados entre dos y cuatro días después de haber reportado el problema técnico.

1.4 Requerimientos del modelo de optimización

El presente estudio busca mejorar el tiempo de respuesta de los técnicos ante las solicitudes de los clientes. Esto se logrará proporcionando a cada técnico una hoja de ruta en la que se presentará la secuencia de clientes a ser visitados con un horario esperado.

Para conseguir este objetivo se elaborará un modelo de optimización que considere:

1. Usar de manera efectiva el tiempo de trabajo de los técnicos.
2. Evitar el traslado innecesario de los técnicos entre clientes muy distantes.

3. Concentrar geográficamente los clientes que serán atendidos por un técnico. De esta manera se logra reducir el tiempo que le toma al técnico trasladarse de un cliente a otro.
4. Tratar en lo posible que los tiempos de servicio y de movilización no excedan el horario de trabajo del técnico.
5. Respetar, en lo posible, los horarios prefijados por los clientes para las visitas de los técnicos.

Capítulo 2

Definiciones y notaciones

Definición 2.1. Red dirigida. [5] [6]

Dado un conjunto V , una *red dirigida* R sobre V es una tripleta $R = (V, A, c)$ donde:

- $A \subseteq V \times V$.
- $c : A \rightarrow \mathbb{R}$

Donde:

- Los elementos de V se llaman *Vértices*
- Los elementos de A se llaman *Arcos*
- c es la función de 'costo' que asigna un número real a cada arco

Definición 2.2. (Vértices inicial y final)

Dada una red $R = (V, A, c)$ y $a = (i, j)$ un arco de A .

- i es el extremo inicial de a (tail)
- j es el extremo final de a (head)
- i es el antecesor de j
- j es el sucesor de i

Definición 2.3. (Arcos paralelos)

Dada una red $R = (V, A, c)$.

Dos arcos que tienen los mismos vértices extremos se denominan *paralelos*

Observación:

Una red $R = (V, A, c)$ se dice *simple* si no tiene arcos paralelos

Observación:

Para hacer una *representación gráfica* de una red $R = (V, A, c)$:

- Se asocia a cada vértice $v \in V$ un punto en el plano
- Todo arco $a \in A$ se asocia con una flecha en el plano que parte desde su extremo inicial hacia el extremo final
- Se coloca sobre cada arco a el valor correspondiente para $c(a)$

Ejemplo 2.1. La Figura 2.1, (p.7) representa esquemáticamente la red $R = (V, A, c)$ definida por los siguientes conjuntos de vértices y arcos.

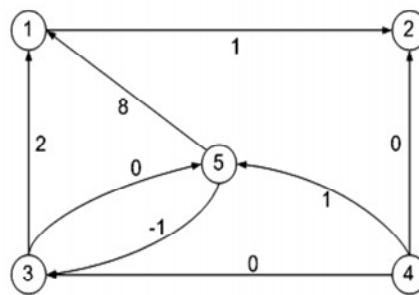


Figura 2.1: Ejemplo de representación gráfica de una red

$$V = \{1, 2, 3, 4, 5\}$$

$$A = \{(1, 2), (3, 1), (3, 5), (4, 2), (4, 3), (4, 5), (5, 1), (5, 3)\}$$

$a \in A$	(1, 2)	(3, 1)	(3, 5)	(4, 2)	(4, 3)	(4, 5)	(5, 1)	(5, 3)
$c(a)$	1	2	0	0	0	1	8	-1

Definición 2.4. (Subred)

Sean $R = (V, A, c)$ una red

Una *subred* de R es una red $R' = (V', A', c)$ tal que $A' \subseteq A$ y $V' \subseteq V$

Ejemplo 2.2. La Figura 2.2, (p.8) muestra una subred $R' = (V', A', c)$ de la red del ejemplo 2.1 definida por:

$$V' = \{1, 2, 3, 5\}$$

$$A' = \{(3, 1), (3, 5), (5, 1)\}$$

Definición 2.5. (Subred inducida)

Sean $R = (V, A, c)$ una red, $V' \subset V$

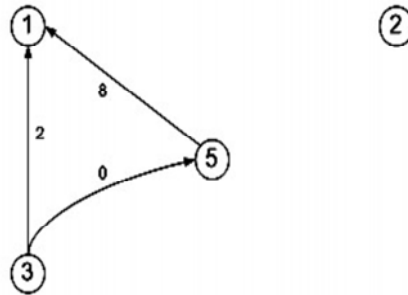


Figura 2.2: Ejemplo de una subred

Una *subred inducida* por V' es una subred $R' = (V', A', c)$ donde A' contiene todas las aristas de A con ambos extremos en V'

Se nota: $R' = R_{V'}$

Ejemplo 2.3. La Figura 2.3, (p.8) muestra una subred inducida $R_{V'}$ de la red del ejemplo 2.1 definida por:

$$V' = \{1, 2, 3, 5\}$$

$$A' = \{(1, 2), (3, 1), (3, 5), (5, 1), (5, 3)\}$$

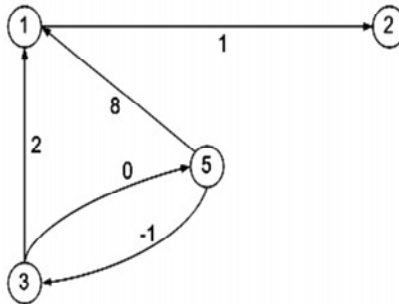


Figura 2.3: Ejemplo de una subred inducida

Definición 2.6. (Lista de adyacencia)

Sea $R = (V, A, c)$ una red

Se define la *lista de adyacencia* de un nodo $i \in V$ como:

$$L(i) = \{j \in V / (i, j) \in A\}$$

Observación:

$$\sum_{i \in V} |L(i)| = |A|$$

Ejemplo 2.4. De la red del ejemplo 2.1 se definen las siguientes listas de adyacencia:

$$L(1) = \{2\}$$

$$L(3) = \{1, 5\}$$

Definición 2.7. (Camino y circuito)

Sean $R = (V, A, c)$ una red y $v, w \in V$.

1. Un *camino* en R de v a w es una secuencia de vértices de V , (v_0, v_1, \dots, v_n) , tal que:

- $v_0 = v$ y $v_n = w$
- $\forall i = 1, \dots, n, \exists a_i \in A, a_i = (v_{i-1}, v_i)$

Por convención se representa también el camino por medio de la secuencia de arcos: (a_1, a_2, \dots, a_n) .

2. Un *camino es simple* cuando los arcos que se encuentran en la secuencia de vértices aparecen una sola vez.

3. Un *circuito* en $v \in V$ es un camino en R de v a v que contiene al menos dos arcos.

Ejemplo 2.5. En la Figura 2.4, (p.9) se pueden apreciar dos caminos y un circuito construidos sobre la red del ejemplo 2.1

Un camino de 4 a 1 puede ser: $(4, 5, 1)$

Otro camino de 4 a 1 puede ser: $(4, 3, 1)$

Un circuito desde 3 puede ser: $(3, 5, 3)$

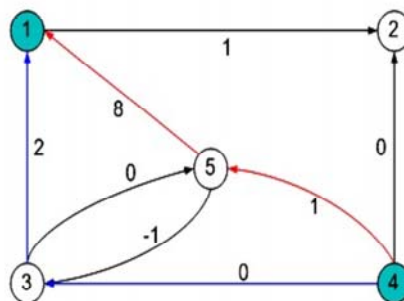


Figura 2.4: Ejemplo de caminos y circuitos en una red

Definición 2.8. (Longitud de un camino)

Dada la red $R = (V, A, c)$

Consideremos el camino de i a j : $P = (a_1, a_2, \dots, a_k)$ con $a_i \in A$, $i = 1, \dots, k$

La *longitud o costo* de un camino P , notada por $c(P)$, se define como

$$c(P) = \sum_{i=1}^k c(a_i)$$

Es decir, la longitud de un camino es igual a la suma de los costos de cada uno de los arcos que lo componen.

Definición 2.9. (Distancia entre vértices)

Dadas la red $R = (V, A, c)$ y dos vértices $i, j \in V$

La *distancia* entre i y j , notada como $d(i, j)$, se define como:

$$d(i, j) = \text{Min}\{c(P) : P \text{ es un camino de } i \text{ a } j\}$$

Capítulo 3

Problema de Enrutamiento de Vehículos (VRP)

3.1 Introducción

El problema de enrutamiento de vehículos (Vehicle Routing Problem, VRP) consiste en optimizar el uso de una flota de vehículos para transportar algún tipo de producto o carga entre un conjunto de depósitos y un conjunto de clientes. Entre las aplicaciones están la recolección de desechos sólidos, el barrido de calles, el ruteo de buses escolares, el ruteo de repartidores, etc.

La red de transporte se modela con un grafo dirigido o no dirigido con costos asociados en sus aristas o arcos. Las rutas deben satisfacer ciertas restricciones particulares del producto que se distribuye y/o de la flota disponible. Entre los objetivos más comunes de optimización, están:

- minimización del costo total de transporte
- minimización del número de vehículos a usarse
- balanceo de las rutas de los camiones
- minimización de penalidades asociadas a clientes parcialmente atendidos
- maximización de la cantidad de carga entregada o recibida, etc.

Los problemas de enrutamiento pueden tener algunas complicaciones que irán más allá del modelo clásico. Las complicaciones típicas se citan a continuación:

1. los costos pueden no ser simétricos
2. la flota de vehículos puede no ser heterogénea, es decir cada vehículo puede tener diferente capacidad

3. el tiempo total de la ruta puede tener valores máximos
4. el tiempo de entrega en el cliente i puede estar restringida a una ventana de tiempo
5. cada vehículo puede estar asignado a diferente depósito
6. el orden de visita a los clientes puede tener alguna prioridad
7. el número de vehículos no es fijo
8. la no visita a ciertos clientes puede estar penalizada
9. algunos productos deben ser entregados con cierta frecuencia (por ejemplo tres veces por semana)
10. en problemas reales la función objetivo puede ser compleja ya que puede incluir términos que dependan de la distancia viajada o del número de vehículos

3.2 Modelos para solucionar el VRP

3.2.1 Modelo basado en arcos

Los modelos de este tipo utilizan variables bivalentes, asociadas a cada arco de la red, para determinar si un camión se desplaza de un cliente a otro. Estos modelos son usados frecuentemente en las versiones básicas del VRP.

Estos modelos se ajustan bien a los casos donde el costo de la solución se expresa como la suma de los costos asociados a los arcos y cuando las restricciones más relevantes tienen que ver con el desplazamiento de un cliente a otro dentro de una ruta.

El modelo más popular para el problema de ruteo, es en el que los vehículos son idénticos y hacen repartos a los clientes desde un depósito central. Para definir el modelo es necesario considerar los siguientes parámetros:

K = número de vehículos

n = número de clientes, los clientes se indexan de 1 a n y se utiliza el 0 para el depósito central

b = la capacidad de cada vehículo

d_i = cantidad despachada en el cliente i medida en la misma unidad que la capacidad del camión

c_{ij} = costo del viaje directo entre los puntos i y j (se asume $c_{ij} \geq 0$ y $c_{ij} = c_{ji}$ para todo ij)

El problema consiste en determinar k rutas de los vehículos, donde una ruta es un ciclo que empieza en el depósito, atraviesa un conjunto de clientes en una secuencia específica y regresa al depósito. Cada cliente debe ser asignado exactamente a una de las K rutas y la cantidad total despachada en el cliente por un vehículo no debe exceder su capacidad b . Las rutas deben ser tomadas de tal manera que minimicen el tiempo total de recorrido.

Para la formulación del modelo es necesario definir las siguientes variables binarias

$$x_{ijk} = \begin{cases} 1 & : \text{ si el vehículo } k \text{ recorre del cliente } i \text{ al cliente } j \\ 0 & : \text{ caso contrario} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & : \text{ si el cliente } i \text{ está en la ruta } k \\ 0 & : \text{ caso contrario} \end{cases}$$

El modelo se presenta a continuación:

$$\min \sum_{i \in V} \sum_{j \in V} \sum_{i=1}^K c_{ij} x_{ijk} \quad (3.1)$$

sujeto a

$$\sum_{k=1}^K y_{ik} = 1, \quad \forall i \in V \setminus \{0\} \quad (3.2)$$

$$\sum_{k=1}^K y_{0k} = K, \quad (3.3)$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = y_{ik}, \quad \forall i \in V, k = 1, 2, \dots, K \quad (3.4)$$

$$\sum_{i \in V} d_i y_{ik} \leq b, \quad \forall k = 1, 2, \dots, K \quad (3.5)$$

$$\sum_{i \in S} \sum_{j \notin S} x_{ijk} \geq y_{hk}, \quad \forall S \subseteq V \setminus \{0\}, h \in S, k = 1, \dots, K \quad (3.6)$$

$$y_{ik} \in \{0, 1\}, \quad \forall i \in V, k = 1, \dots, K \quad (3.7)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall i, j \in V, k = 1, \dots, K \quad (3.8)$$

Donde:

- 3.1: es la función objetivo de minimización
- 3.2: es la restricción que garantiza que cada cliente sea visitado exactamente una vez por algún camión
- 3.3: implica que deben salir exactamente K camiones del depósito
- 3.4: implica que si el cliente i es visitado por el camión k , entonces este camión debe 'llegar' y 'salir' del cliente i
- 3.5: implica que si el cliente i es visitado por el camión k , entonces su demanda d_i no debe exceder la capacidad del camión
- 3.6: garantiza que no se produzcan subciclos en las k rutas

Observación:

Se han hecho extensiones y adaptaciones al modelo, algunas de las cuales son:

- Cuando el grafo no es completo se completan los arcos que faltan y se ponen costos asociados grandes (infinitos). Cuando el número de arcos es muy grande se debe tomar ventaja de la forma dispersa de la matriz asociada.
- Otra modificación constituye el reemplazo del depósito simple con uno para cada vehículo.
- También es factible asignar a cada vehículo una capacidad diferente.

3.2.2 Modelo basado en rutas

Estos modelos tienen un número exponencial de variables binarias, cada una asociada con una ruta factible diferente. El VRP de este estilo es formulado como un problema de particionamiento [4] en el que se busca determinar una colección de rutas al mínimo costo, que sirva a cada cliente una sola vez, y que posiblemente, satisfaga otras restricciones.

La principal ventaja de este tipo de modelo es que permite costear cada ruta de varias formas e incluso puede considerar restricciones individuales para cada ruta.

Para definir el modelo se consideran las siguientes variables y parámetros:

- Si $\Gamma = \{G_1, G_2, \dots, G_q\}$ son todos los circuitos, cada uno correspondiente a una ruta factible. Sea $q = |\Gamma|$
- Todo circuito G_i tiene asociado un costo c_i

- Se denota con a_{ij} el coeficiente bivalente que toma el valor de 1 cuando el vértice i es visitado (o cubierto) por la ruta G_j y 0 en caso contrario.
- Se denota con x_j ($j = 1, \dots, q$) la variable bivalente que toma el valor de 1 cuando el circuito G_j es tomado como solución óptima.

Formulación matemática:

$$\min \sum_{j=1}^q c_j x_j \quad (3.9)$$

sujeto a

$$\sum_{j=1}^q a_{ij} x_j = 1, \quad \forall i \in V \setminus \{0\} \quad (3.10)$$

$$\sum_{j=1}^q x_j = K, \quad (3.11)$$

$$x_j \in \{0, 1\}, \quad \forall j = 1, \dots, q \quad (3.12)$$

Donde:

3.9: es la función objetivo de minimización

3.10: exige que cada cliente i sea cubierto por uno y solo un circuito

3.11: exige que se tomen exactamente K circuitos

Para resolver este tipo de modelos se utiliza la descomposición de Dantzig-Wolfe [7] la cual permite dividir el problema en partes más fáciles de resolver.

Esta descomposición es un procedimiento clásico para resolver problemas lineales con determinadas características de bloque. Relajando ciertas restricciones permite dividir el problema en varios problemas independientes más pequeños.

Para problemas como el VRP, en donde existen un gran número de columnas (una columna para cada ruta), se puede aplicar el método de generación de columnas [7].

3.3 Variantes del VRP

3.3.1 VRP con ventanas de tiempo

Este modelo es una extensión del modelo clásico de VRP que considera lo siguiente:

- A cada cliente i se le asocia un intervalo de tiempo $[a_i, b_i]$ llamada *ventana de tiempo*.
- A cada cliente se le asocia también un tiempo de servicio s_i .
- En el caso de arribo temprano el vehículo debe esperar hasta el instante a_i que es el tiempo cuando el servicio puede empezar.
- Para cada cliente i , el servicio empieza dentro de la ventana $[a_i, b_i]$ y el vehículo se demora en atenderlo s_i instantes.

3.3.2 VRP con recargas

Es una extensión del modelo clásico de VRP que considera lo siguiente:

- Los vértices (sin el depósito) se dividen en dos grupos (subconjuntos). El primero $L = \{1, 2, \dots, n\}$ correspondiente a los clientes que solo demandan y el segundo conjunto $B = \{n + 1, n + 2, \dots, n + m\}$ de clientes que proveen de producto.
- Existe precedencia entre los vértices de L y B . Los de L deben ser servidos primero.
- La demanda total de un cliente de L no debe exceder la capacidad del vehículo.
- En cada circuito todos los clientes de L preceden a los de B .
- No están permitidos circuitos solo de vértices de B .

3.3.3 VRP con cargas y descargas

Se basa en el modelo clásico de VRP y considera adicionalmente:

- A cada cliente i se le asocian dos cantidades:
 1. d_i : la cantidad demandada de producto, y
 2. p_i : la cantidad a ser recogida del cliente.
- Para cada cliente i , se asocian dos vértices:
 1. un vértice O_i que es el vértice de origen de la demanda, y
 2. D_i que es un vértice de recarga.

- Se asume que en cada cliente primero se descarga y luego se carga.
- La carga total del vehículo a través de la ruta no debe exceder su capacidad.
- Para cada cliente i , el cliente O_i , cuando es diferente del depósito, debe ser servido en el mismo circuito y antes del cliente i .
- Para cada cliente i , el cliente D_i , cuando es diferente del depósito, debe ser servido en el mismo circuito y después del cliente i .

3.4 Heurísticas clásicas para el VRP

3.4.1 Introducción

Estas heurísticas se pueden clasificar en:

Clásicas. Tienen un mayor desarrollo, son 'fáciles' de implementar y sus soluciones son aceptables. Se encuentran implementadas en la mayoría de paquetes comerciales. Las heurísticas clásicas se dividen en tres categorías:

- *Heurísticas Constructivas.* Se basan en la construcción de soluciones factibles mirando siempre la función objetivo.
- *Heurísticas de dos-fases.* Dividen el problema en dos componentes: agrupando los vértices en rutas factibles y la ruta actual. Estas heurísticas se dividen en:

Agrupe primero - rutee luego. En este caso, se forman primero los grupos de vértices y luego se encuentran rutas factibles en cada uno.

Rutee primero - agrupe luego. En este caso, se construye un ciclo en todos los vértices y luego se segmentan en rutas factibles.

Metaheurísticas. Para su desarrollo requieren de procedimientos más sofisticados, dan mejores resultados pero requieren de mejores prestaciones computacionales.

Métodos de mejoramiento. Estos métodos buscan mejorar una solución factible haciendo una secuencia de cambios de vértices y arcos.

3.4.2 Métodos Constructivos

Algoritmo de los ahorros de Clarke and Wright

Es el algoritmo más usado para problemas VRP. Se basa en la noción de ahorros. Cuando dos rutas $(0, \dots, i, 0)$ y $(0, j, \dots, 0)$ pueden ser unidas en una sola $(0, \dots, i, j, \dots, 0)$ se genera un ahorro en distancia igual a $s_{ij} = c_{i0} + c_{0j} - c_{ij}$. Los pasos a seguir son:

Paso 1. (Cálculo de ahorros)

Calcular $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ para $i, j = 1, \dots, n$ y $i \neq j$

Ordenar los ahorros de manera decreciente

Versión paralela

Paso 2. (La mejor unión factible)

Partiendo de la lista de ahorros. Hacer:

Dado s_{ij} determinar si existen dos rutas, una conteniendo el arco $(0, j)$ y otra conteniendo el arco $(i, 0)$ que puedan ser factibles de unir. Si es posible, borrar $(0, j)$ y $(i, 0)$ e introducir (i, j) .

Versión secuencial

Paso 2. (Extensión de la ruta)

Consideremos cada ruta $(0, i, \dots, j, 0)$. Determinar el primer ahorro s_{ki} o s_{jl} factibles que puedan ser usadas para unir la ruta con otra conteniendo el arco o arista $(k, 0)$ o $(0, l)$.

Implementar la unión y repetir la operación para la ruta actual.

Si no hay una ruta factible, considere una nueva.

Si no hay mas rutas factibles, parar.

Observación:

La versión paralela arroja mejores resultados que la versión secuencial.

Mejoras al Algoritmo de Clarke and Wright

El algoritmo original presenta buenas rutas al inicio pero menos interesantes al final, incluyendo rutas circulares. Para remediar esto Gaskel y Yellow propusieron un ahorro generalizado definido como $s_{ij} = c_{i0} + c_{0j} + \lambda c_{ij}$, donde λ es una parámetro de la forma

de la ruta. Mientras más grande es λ más énfasis se pone en la distancia entre los vértices que se conectarán. Se reportaron buenas soluciones con valores de λ de 0.4 o 1.0, considerando el número de rutas y la longitud total de la solución.

El algoritmo de Clarke y de Wright puede también ser caro en cuanto a cálculos, ya que todos los ahorros deben ser computados, ser almacenados, y ser clasificados. Varias mejoras han sido propuestas para acelerar cómputos y para reducir requisitos de la memoria. La mayoría de estos trabajos ocurrieron en los años 70 y 80. Estas mejoras son solo apreciables hoy en día solamente para los casos muy grandes (más de 1000 vértices).

Nelson y otros investigaron estructuras de datos más complejas basadas en montones para evitar las limitaciones de almacenaje y así obtener operaciones de actualización más eficientes.

3.4.3 Heurísticas de inserción secuencial

Existen dos algoritmos basados en inserciones secuenciales. Ambos se aplican a los problemas con un número de vehículos no especificado. El primer algoritmo elaborado por Mole y Jameson, expande una ruta a la vez. El segundo, propuesto por Christofides, Mingozzi, y Toth aplica en cambio procedimientos de construcción de rutas secuenciales y paralelos.

Heurística de inserción secuencial de Mole y Jameson

Utilizamos dos parámetros: λ y μ para expandir la ruta en construcción

$$\alpha(i, k, j) = c_{ik} + c_{kj} - \lambda c_{ij}$$

$$\beta(i, k, j) = \mu c_{0k} - \alpha(i, k, j)$$

Paso 1: Inicialización de la ruta

Si todos los vértices pertenecen a una ruta, parar.

Si no, construya una ruta $(0, k, 0)$, donde k es un vértice cualquiera fuera de ruta .

Paso 2: Vértice siguiente

Para cada vértice k calcule el costo de inserción factible $\alpha^*(i_k, k, j_k) = \min\{\alpha(r, k, s)\}$ para todos los vértices adyacentes r y s de una ruta, donde

i_k y j_k son los dos vértices sobre los que se calcula α^* . Si no hay inserción factible, vaya al paso 1. De otra manera, el mejor vértice k^* a insertar en la ruta es el vértice que descansa sobre $\beta^*(i_k, k^*, j_k) = \max\{\beta(i_k, k, j_k)\}$ sobre todos los vértices k fuera de ruta.

Inserte k^* entre i_{k^*} y j_{k^*}

Paso 3 Optimización de la ruta

Optimice la ruta actual por medio de un procedimiento 2-opt ó 3-opt, y vaya al paso 2.

Observación:

Varias reglas de inserción son gobernadas por los dos parámetros λ y μ . Por ejemplo, si $\lambda = 1$ y $\mu = 0$, el algoritmo inserta el vértice que tiene la distancia adicional mínima. Si $\lambda = \mu = 0$, el vértice que se insertará corresponde a la suma más pequeña de distancias entre dos vecinos.

Si $\mu = \infty$ y $\lambda > 0$, el vértice más distante del depósito será insertado.

Heurística de inserción secuencial de Christofides, Mingozzi, y Toth.

Christofides, Mingozzi y Toth desarrollaron una inserción de dos-fases algo más sofisticada que utiliza también dos parámetros λ y μ .

Fase 1. Construcción secuencial de la ruta

Paso 1. Primera ruta

Fije el índice de la ruta $k = 1$

Paso 2. Costes de la inserción

Seleccione cualquier vértice fuera de ruta para inicializar la ruta k . Para cada vértice i fuera de ruta, calcule $\delta_i = c_{0i} + \lambda c_{ii_k}$.

Paso 3. Inserción del vértice

Sea $\delta_{i^*} = \min_{i \in S_k} \{\delta_i\}$, donde S_k es el conjunto de vértices fuera de ruta que son factibles de insertar en la ruta k .

Optimize la ruta k usando un algoritmo 3-opt

Repita 3 hasta que ningún otro vértice pueda ser asignado a la ruta k .

Paso 4. Próxima ruta

Si todos los vértices han sido insertados en rutas, parar.

Caso contrario, $k = k + 1$ y vaya a 2

Fase 2. Construcción de la ruta paralela

Paso 5. Inicialización de la ruta

Inicialice k rutas $R_t = (0, i_t, 0)$ para $t = 1, \dots, k$, donde k es el número de rutas obtenidas en la fase 1.

Sea $J = \{R_1, \dots, R_k\}$

Paso 6. Asociación de costos

Para cada vértice i no asociado a ninguna ruta y para cada ruta factible $R_t \in J$, calcule $\varepsilon_{ti} = c_{0i} + \mu c_{ii}$ y $\varepsilon_{t^*i} = \min_t \{\varepsilon_{ti}\}$. Asocie el vértice i con la ruta R_t y repita 6 hasta que todos los vértices estén asociados a una ruta.

Paso 7. Inserción de costos

Tome una ruta R_t y sea $J = J \setminus \{R_t\}$. Para cada vértice i asociado a la ruta R_t , calcule $\varepsilon_{t'i} = \min_{R_t} \{\varepsilon_{ti}\}$ y $\tau_i = \varepsilon_{t'i} - \varepsilon_{ti}$

Paso 8. Inserción de vértices

Inserte en la ruta R_t el vértice i^* satisfaciendo $\tau_{i^*} = \max_{i \in S_t} \{\tau_i\}$. Optimice R_t . Repita 8 hasta que no se puedan insertar más vértices en la ruta R_t

Paso 9. Test de parada

Si $|J| \neq \emptyset$ vaya a 6.

Caso contrario, si todos los vértices están enrutados, parar.

Si quedan vértices sin ruteo, cree nuevas rutas con el paso 1 de la fase 1.

Observación:

Se ha determinado que este último algoritmo es superior al de Mole y Jameson

3.4.4 Métodos de dos fases

Los métodos de dos fases primero empaquetan los vértices en grupos llamados 'clusters' y luego resuelven individualmente un problema de ruteo en cada uno. Entre los principales algoritmos podemos mencionar [11]:

- algoritmo de barrido

- algoritmo de Fisher y Jakumar
- algoritmo Bramel y Simchi-Levi
- algoritmos pétalo

3.4.5 Heurísticas de mejoramiento

Operan en cada ruta tomada por separado o en varias rutas a la vez. En el primer caso, cualquier heurística mejorada para el TSP puede ser aplicada. En el segundo caso, se pueden desarrollar procedimientos que aprovechen la estructura de multiruta del VRP.

Mejoramiento de rutas simples

Muchos procesos mejorados para el TSP se pueden describir en términos del mecanismo $\lambda - opt$. Aquí, los λ arcos se retiran del tour y los λ segmentos restantes son reconectados de todas las maneras posibles. Si se identifica cualquier reconexión provechosa (la primera o la mejor), se la ejecuta.

El procedimiento se detiene en un mínimo local cuando ninguna otra mejora puede ser obtenida. La λ -optimalidad de la solución puede obtenerse en $O(n^\lambda)$ tiempo.

Mejoramiento en multi-rutas

Van Breedam considera las operaciones de mejoramiento como casos especiales de intercambios $2 - opt$, y provee de análisis computacionales en problemas de prueba. Kindervater y Savelsbergh definen operaciones similares y realizan experimentos sobre todo en el contexto del VRP con ventanas de tiempo.

El análisis de Van Breedam puede ser resumido en las cuatro operaciones siguientes:

1. Cruce de secuencias. Dos secuencias (o cadenas) de vértices son intercambiadas cruzando dos arcos de dos rutas diferentes.
2. Intercambio de secuencias. Dos cadenas de a lo más k vértices se intercambian entre dos rutas

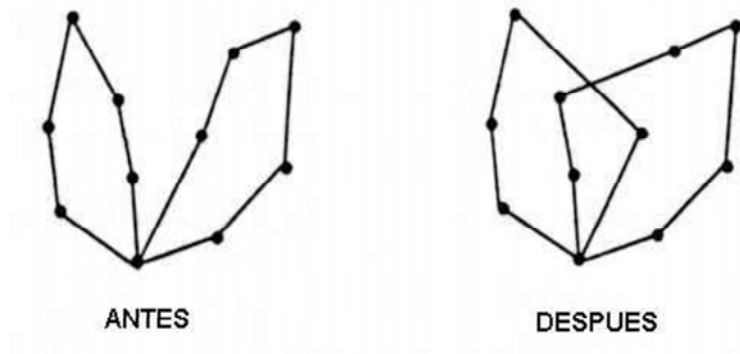


Figura 3.1: Cruce de secuencias

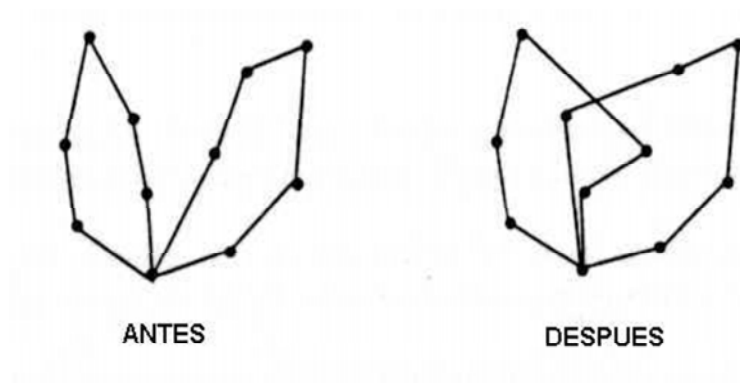


Figura 3.2: Intercambio de secuencias

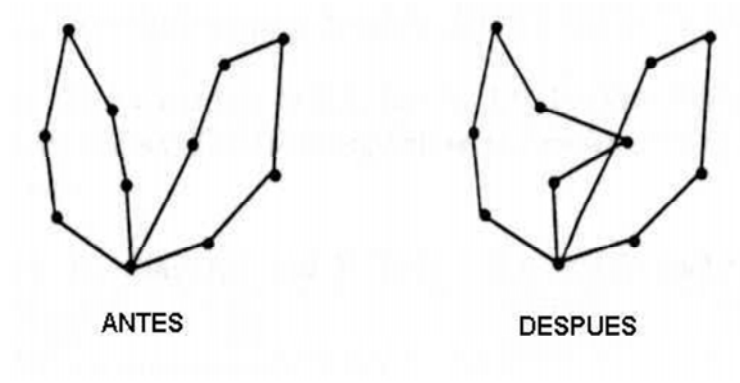


Figura 3.3: Relocalización de secuencias

3. Relocalización de secuencias. Una cadena de a lo más k vértices se mueve de una ruta a otra, típicamente con $k = 1$ ó 2
4. Mezcla de secuencias. Se selecciona el mejor movimiento de los intercambios $2 - opt$ y $3 - opt$

Observaciones:

- Es mejor iniciar la búsqueda con una buena solución en lugar de una solución pobre, en términos de la calidad final de la solución y del tiempo de cálculo.
- Los movimientos del segundo intercambio parecen ser los mejores. Esto se ha confirmado aplicándolos en mejoras locales en recocido simulado y búsquedas tabú.
- Los intercambios de tipo 2 son mejores que los de tipo 1, pero el tiempo de cálculo se cuadriplica.
- Las comparaciones entre estas heurística no siempre son fáciles de hacer debido especialmente a la manera como son implementadas en la práctica.
- Está claro que, en términos de la calidad de la solución, las heurísticas clásicas basadas en construcción no compiten con las metaheurísticas. Sin embargo, varios de los métodos presentados se pueden adaptar fácilmente a otras variantes del VRP y son fáciles de implementar. Esto explica su uso extendido en el software comercial.
- El algoritmo de Clarke y de Wright sigue siendo probablemente el método más popular en la práctica.

Capítulo 4

Modelo de P.E. para el problema de rutas

4.1 Conjuntos necesarios

Sea $\tilde{C} = \{c_1, c_2, \dots, c_{n_c}\}$ el conjunto de clientes localizados en la ciudad de Quito y $C \subseteq \tilde{C}$ el conjunto de clientes a ser atendidos en un día específico.

Se definen como 0 y -1 los vértices de salida y llegada, respectivamente, de los técnicos.

Se define de manera similar el conjunto de técnicos $\tilde{K} = \{k_1, k_2, \dots, k_{n_K}\}$ que dan servicio en la ciudad de Quito y sus alrededores y el conjunto $K \subseteq \tilde{K}$ como el conjunto de técnicos que prestarán sus servicios en un día específico.

Para modelizar la red vial de la ciudad se requiere definir el conjunto de puntos de tránsito $P = \{p_1, p_2, \dots, p_{n_p}\}$

4.2 Red auxiliar

4.2.1 Definición

La red auxiliar $\tilde{R} = (\tilde{V}, \tilde{A}, \tilde{c})$ se construye en función de los conjuntos y función de costos descritos a continuación:

Conjunto de vértices

El conjunto de vértices de la red auxiliar se conforma por los clientes a ser atendidos en un día específico C , los vértices de salida y llegada (0 y -1) y los puntos de tránsito P .

Conjunto de arcos

Se determinaron tres tipos de arcos:

1. Los arcos entre puntos de tránsito que representan la circulación respetando barreras físicas y restricciones legales.
2. Los arcos que permiten a los clientes y a los técnicos alcanzar un punto de tránsito.
3. Los arcos que permiten alcanzar un cliente y la oficina principal desde un punto de tránsito.

Función de costo

La función de costo se define como el tiempo que le tomaría a un vehículo motorizado ir del extremo inicial al final de cada arco.

4.2.2 Utilidad

La red auxiliar se construye para poder determinar los tiempos más cortos entre cada uno de los clientes y la oficina central. Para este cálculo se utilizó el algoritmo de Dijkstra [4] el cual funciona muy bien para este tipo de redes.

4.3 Red principal

4.3.1 Definición

La red principal $R = (V, A, c)$ se construye en función de los conjuntos y función de costos descritos a continuación:

Conjunto de vértices

El conjunto de vértices de la red principal se conforma por los clientes a ser atendidos en un día específico C y los vértices de salida y llegada (0 y -1).

Conjunto de arcos

El conjunto de arcos A se determina como $A = V \times V$ (red completa)

Función de costo

La función de costo se define como el tiempo del camino más corto en el que un vehículo motorizado iría del extremo inicial al final de cada arco.

El tiempo del camino más corto entre los clientes y la oficina central se determinó en la red auxiliar. En la red principal solo se asocia a cada arco este valor.

Ejemplo 4.1. En este ejemplo se toma el arco (i, j) que va del vértice i al j ($i, j \in V$). La función de costo asociada a este arco $c(i, j)$ (en color rojo) es el menor tiempo que le tomaría a un vehículo motorizado desplazarse de i a j . Para encontrar este valor se debe calcular el camino más corto entre los vértices i y j sobre la red auxiliar \tilde{R} (arcos azules). Figura 4.1, (p.27)



Figura 4.1: Ejemplo del costo de un arco en la red R

4.4 Consideraciones de tiempo

Es necesario definir:

Tiempo de tránsito (t_{ij}). Es el tiempo que le toma a un vehículo motorizado desplazarse desde un vértice a otro en la red principal

Tiempo de servicio (τ). Es el tiempo promedio que le tomará a un técnico para solucionar cualquier problema en un cliente. De acuerdo a datos de la empresa este

valor es de 30 minutos (0.5 horas)

Ventana de tiempo $[a_i, b_i]$. Es el intervalo de tiempo en el que el cliente i puede recibir al técnico.

4.5 Compatibilidad

Diremos que el arco (i, j) es *compatible* si se respetan las restricciones de tiempo que permitirán a cualquier técnico ir al cliente j directamente después de atender al cliente i , esto es:

$$a_i + \tau + t_{ij} \leq b_j$$

Se define el conjunto de arcos compatible \hat{A} como:

$$\hat{A} = \{(i, j) : (i, j) \text{ es compatible} \}$$

4.6 Variables de decisión

Para la formulación del problema como un programa entero definimos dos tipos de variables:

1. A todo arco $(i, j) \in \hat{A}$, asociamos una variable binaria de flujo x_{ijk} que indica si (i, j) es usada en la ruta del técnico k o no.
2. Para cualquier nodo $i \in V \setminus \{0\}$ visitada en esta ruta, T_i^k es el tiempo en el cual empieza el servicio en el cliente i .

4.7 Modelo de optimización preliminar MOP

El modelo se formula de la siguiente manera:

$$\min \sum_{k \in K} \sum_{(i,j) \in \hat{A}} t_{ij} x_{ijk} \quad (4.1)$$

sujeto a

$$\sum_{k \in K} \sum_{(i,j) \in \hat{A}} x_{ijk} = 1, \quad \forall i \in V \setminus \{0, -1\} \quad (4.2)$$

$$\sum_{(0,j) \in \hat{A}} x_{0,j,k} = 1, \quad \forall k \in K \quad (4.3)$$

$$\sum_{(i,-1) \in \hat{A}} x_{i,-1,k} = 1, \quad \forall k \in K \quad (4.4)$$

$$\sum_{(i,j) \in \hat{A}} x_{ijk} - \sum_{(j,i) \in \hat{A}} x_{jik} = 0, \quad \forall i \in V \setminus \{0, -1\}, \forall k \in K \quad (4.5)$$

$$x_{ijk}(T_i^k + t_{ij} + \tau - T_j^k) \leq 0, \quad \forall i, j \in V \setminus \{0, -1\}, \forall k \in K \quad (4.6)$$

$$a_i \leq T_i^k \leq b_i, \quad \forall i \in V \setminus \{0\}, \forall k \in K \quad (4.7)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall (i, j) \in \hat{A}, \forall k \in K \quad (4.8)$$

La función de minimización 4.1 busca minimizar el tiempo total de traslado de todos los técnicos que van a dar servicio a los clientes.

La restricción 4.2 refleja que cada cliente debe ser visitado exactamente una vez por un técnico. Se excluyen los vértices 0 y -1 que corresponden a la oficina central.

Las restricciones 4.3 y 4.4 son restricciones que obligan a todos los técnicos a salir y entrar al vértice correspondiente a la oficina central.

La restricción 4.5 es una restricción de flujo que implica que si un técnico llega a un cliente debe salir de él. Se excluyen nuevamente los vértices 0 y -1 .

Y por último las restricciones 4.6 y 4.7 aseguran el respeto a las ventanas de tiempo.

La desigualdad en 4.6 es estricta si el técnico llega antes de la hora fijada por el cliente y tiene que esperar antes de empezar el servicio.

Hay que tomar en cuenta que la restricción 4.6 es no lineal por lo que será necesario buscar la manera de hacerla lineal. Ventajosamente x_{ijk} es bivalente, por tanto la restricción 4.6 puede ser reemplazada por

$$T_i^k + \tau + t_{ij} - T_j^k \leq (1 - x_{ijk})M_{ij} \quad \forall (i, j) \in \hat{A}, \quad \forall k \in K$$

donde M_{ij} es una constante suficientemente grande. Debido a 4.7 es suficiente con

tomar

$$M_{ij} = \max \{b_i + t_{ij} + \tau - a_j, 0\} \quad \forall (i, j) \in \hat{A}$$

4.8 Modelo de programación entera MPE

En base a consideraciones realizadas podemos formular el modelo MOP como un modelo de programación entera

$$\min \sum_{k \in K} \sum_{(i,j) \in \hat{A}} t_{ij} x_{ijk} \quad (4.9)$$

sujeto a

$$\sum_{k \in K} \sum_{(i,j) \in \hat{A}} x_{ijk} = 1, \quad \forall i \in V \setminus \{0, -1\} \quad (4.10)$$

$$\sum_{(0,j) \in \hat{A}} x_{0,j,k} = 1, \quad \forall k \in K \quad (4.11)$$

$$\sum_{(i,-1) \in \hat{A}} x_{i,-1,k} = 1, \quad \forall k \in K \quad (4.12)$$

$$\sum_{(i,j) \in \hat{A}} x_{ijk} - \sum_{(j,i) \in \hat{A}} x_{jik} = 0, \quad \forall i \in V \setminus \{0, -1\}, \forall k \in K \quad (4.13)$$

$$x_{ijk} M_{ij} + T_i^k + \tau + t_{ij} - T_j^k \leq M_{ij}, \quad \forall i, j \in V \setminus \{0\}, \forall k \in K \quad (4.14)$$

$$a_i \leq T_i^k \leq b_i, \quad \forall i \in V \setminus \{0\}, \forall k \in K \quad (4.15)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall (i, j) \in \hat{A}, \forall k \in K \quad (4.16)$$

Las consideraciones hechas en la sección anterior tanto en la función objetivo como en las restricciones se mantienen para este modelo.

4.9 Modelo con ventanas de tiempo suaves MVTs

El modelo anterior (MPE) resultó ser computacionalmente muy difícil de resolver para ciertas instancias prácticas del problema debido posiblemente al alto número de variables bivalentes y a las restricciones concernientes a las ventanas de tiempo. Por este motivo, consideramos un modelo alternativo donde relajamos las restricciones 4.6 y

las incorporamos como una nueva componente en la función objetivo.

$$\min \sum_{k \in K} \sum_{(i,j) \in \hat{A}} t_{ij} x_{ijk} + Q \sum_{k \in K} \sum_{(i,j) \in \hat{A}} (x_{ijk} M_{ij} + T_i^k + \tau + t_{ij} - T_j^k - M_{ij}) \quad (4.17)$$

sujeto a

$$\sum_{k \in K} \sum_{(i,j) \in \hat{A}} x_{ijk} = 1, \quad \forall i \in V \setminus \{0, -1\} \quad (4.18)$$

$$\sum_{(0,j) \in \hat{A}} x_{0,j,k} = 1, \quad \forall k \in K \quad (4.19)$$

$$\sum_{(i,-1) \in \hat{A}} x_{i,-1,k} = 1, \quad \forall k \in K \quad (4.20)$$

$$\sum_{(i,j) \in \hat{A}} x_{ijk} - \sum_{(j,i) \in \hat{A}} x_{jik} = 0, \quad \forall i \in V \setminus \{0, -1\}, \forall k \in K \quad (4.21)$$

$$a_i \leq T_i^k \leq b_i, \quad \forall i \in V \setminus \{0\}, \forall k \in K \quad (4.22)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall (i, j) \in \hat{A}, \forall k \in K \quad (4.23)$$

Este modelo pudo ser resuelto con el solver GLPK manteniendo las variables binarias x_{ijk} en tiempos razonables (1 y 4 minutos). Desventajosamente debido a la relajación se producen subciclos en las rutas de los técnicos lo que hace necesario incorporar alguna heurística de inserción para conformar un solo ciclo por cada técnico.

4.10 Heurística de Inserción

Cuando se trabaja con el modelo relajado no hay la garantía de crear las rutas completas (un solo ciclo) para cada técnico, sino que se crean varios subciclos que se deben consolidar en uno solo para cada técnico.

Para hacer esto se utiliza una heurística de inserción, la misma que va incorporando los clientes al ciclo principal del técnico. De los algoritmos de inserción revisados en el capítulo 3 se escogió el de Mole and Jameson¹ debido principalmente a que:

- permite la inserción de clientes fuera de la ruta del técnico de manera progresiva hasta completar su jornada de trabajo. Una vez completado este tiempo se procede de manera similar con el siguiente técnico.

¹Ver 3.4.3

- permite tener un número variable de técnicos. Estos van apareciendo de acuerdo a la necesidad de visitar a los clientes
- permite considerar las ventanas de tiempo haciendo factible o no la inserción de un cliente en determinada posición de la ruta

Luego de varias pruebas de esta heurística sobre los modelos de 15 clientes se llegó a la conclusión que los mejores valores para λ y μ eran de 0.1 y 2.3 respectivamente. Todos los ejemplos de prueba se implementaron con estos valores.

Capítulo 5

Implementación computacional

5.1 Localización geográfica de vértices y arcos

5.1.1 Localización de los clientes

Para la ubicación geográfica de los clientes en el mapa de la ciudad de Quito se registró por medio de un GPS la posición de cada uno de ellos en las coordenadas internacionales UTM (*Universal Traverse Mercator*). Estas coordenadas se almacenaron en una base de datos conjuntamente con otros datos del cliente: dirección, razón social, RUC, teléfonos, etc. La información geográfica de los clientes permite su ubicación sobre el mapa digital de la ciudad de Quito para fines ilustrativos y de análisis. Ver Figura 5.1, (p.34)

En la base de datos se registraron los vértices clientes utilizando índices enteros negativos.

5.1.2 Localización de la oficina central

Para la ubicación geográfica de la oficina central en el mapa de la ciudad de Quito se utilizó un GPS el cual registró su posición en las coordenadas UTM.

Obligatoriamente los técnicos deben estar presentes en la oficina central a primera hora en la mañana pues a esta hora el personal del departamento técnico les provee de:

- La hoja ruta, es decir el listado de los clientes que deben visitar.
- Documentos varios: facturas, mapas de instalación, proformas, etc.
- Materiales: sistemas de seguridad, repuestos, partes y piezas de alarmas, etc.

De la misma manera, el personal técnico debe reportarse al finalizar la jornada para:



Figura 5.1: Localización de los clientes en el mapa de Quito

- Entregar los estudios de localización de nuevos equipos para ampliar los sistemas de seguridad (nuevas zonas).
- Entregar los resultados de los trabajos realizados en cada cliente visitado.
- Entregar los documentos firmados por los clientes.
- Notificar sobre novedades o quejas suscitadas en los clientes.

En la base de datos se registró la oficina central con dos vértices 0 y -1.

5.1.3 Puntos de tránsito

Los puntos de tránsito son vértices que se construyen sobre el mapa digital de la ciudad de Quito y son determinados por el cruce de dos o más avenidas y/o calles principales. Figura 5.2, (p.35)

Se emplean para modelizar las restricciones físicas o de tipo legal para los desplazamientos sobre la red vial.

El número de puntos que se construyen en cada cruce depende del tipo de cruce y el nivel de detalle con el que se quiere representar la circulación¹.

¹Para obtener información de los detalles de la construcción de puntos de tránsito ver el apéndice B.1



Figura 5.2: Ejemplo de puntos de tránsito

En la base de datos se registraron los vértices puntos de tránsito utilizando índices enteros positivos.

5.1.4 Arcos

Red auxiliar

En la red auxiliar $\tilde{R} = (\tilde{V}, \tilde{A}, \tilde{c})$ los arcos se determinan entre dos puntos de tránsito $p_i, p_j \in P$ cuando es factible un movimiento entre ellos. Esto se determina por el hecho de que un vehículo motorizado se pueda desplazar directamente desde el punto p_i al punto p_j respetando las barreras arquitectónicas, el sentido de las calles y avenidas y las señales de tránsito². El costo asociado a cada arco es el tiempo que le toma al vehículo ir del un extremo al otro, este cálculo se hace dividiendo la distancia euclídea entre los dos puntos de tránsito para la velocidad promedio de un vehículo motorizado en la ciudad (10 km/h).

En la red auxiliar también es necesario incorporar los arcos que salen de los clientes y oficina central hacia el punto de tránsito más cercano y también los arcos que causan el efecto contrario. El costo asociado en este caso se calcula en base a la distancia euclídea dividida para la velocidad de un caminante (4 km/h)

²Para obtener mayor información de cómo determinar movimientos factibles entre puntos de tránsito ver el apéndice B.2

Red principal

En el caso de la red principal, los arcos se determinan haciendo $A = V \times V$ y sus costos asociados se determinan calculando para cada par de vértices el camino más corto (en tiempo) sobre la red auxiliar y el tiempo total corresponderá al costo asociado al arco entre ese par de vértices.

5.2 Características del hardware y software

5.2.1 Hardware

La presente aplicación, base de datos y análisis de resultados se ejecutaron en un computador con las siguientes características:

- Procesador Pentium IV con 1.7Ghz de velocidad
- Memoria RAM de 650Mb
- Disco duro IDE de 60Gb

5.2.2 Software

Con respecto a las aplicaciones se utilizaron las siguientes:

- Windows 2000 Server como sistema operativo
- Base de datos en Access 2000 [15]
- PowerDesigner 9.0 para el diseño de la base de datos
- Visual Basic 6.0 para la interfase con la base de datos
- Dev-C++ para la red y caminos más cortos con sus librerías estándar [12]
- GLPK para la solución del problema lineal y mixto [13]
- ArcView 3.2 para el manejo de los datos geográficos y mapas digitales

5.3 Base de datos de la red de Quito y clientes

5.3.1 Red de Quito

Para la creación de la red de la ciudad de Quito se trabajó con el mapa digital de la ciudad, sobre el cual se construyeron los vértices y arcos en las posiciones geográficas de calles e intersecciones (Figura 5.3, (p.37)). Se respetaron los sentidos de calles y avenidas así como las barreras arquitectónicas y prohibiciones de tránsito. Estos datos se registraron en una base de datos de Access (Figura 5.5, (p.41))



Figura 5.3: Vértices y arcos sobre el mapa digital

Vértices

Previo al ingreso de la información se cargó el mapa digital de la ciudad de Quito en el programa ArcView 3.2. Esta herramienta permite visualizar las coordenadas en UTM de cualquier punto específico del mapa.

De esta manera fue posible determinar las posiciones de los vértices (puntos de tránsito) y registrarlos directamente en la base de datos. A cada vértice se le identificó con un número entero positivo único.

Se registraron 1538 puntos de tránsito en la base de datos.

Arcos

Los arcos fueron ingresados directamente en la base de datos. Cada arco está representado por un par ordenado de vértices compatibles (i, j) , y su costo se definió como la distancia euclidiana (en metros) obtenida de las coordenadas UTM.

De esta manera se registraron 2152 arcos.

5.3.2 Localización de clientes

La ubicación de los clientes se utilizó utilizando un GPS³ para registrar sus coordenadas geográficas .

Se identificó a cada cliente con un número negativo único. Además se determinaron los puntos de tránsito más cercanos a cada cliente desde los que puede entrar o salir hacia o desde la red auxiliar.

La tabla de clientes registra la siguiente información:

- Coordenadas UTM del cliente
- Vértice más cercano a la Red Digital de Quito (RDQ) que le servirá como punto de partida para desplazarse dentro de la red.
- Distancia desde el vértice anterior al cliente (en metros)
- Vértice más cercano desde donde cualquier desplazamiento dentro de la RDQ puede alcanzar al cliente
- Distancia desde el vértice anterior al cliente (en metros)

Ver Figura 5.4, (p.39)

³Geographical Positioning System



Figura 5.4: Clientes sobre el mapa digital

5.3.3 Estructura de la base de datos

La base de datos se estructuró alrededor de tres tablas (entidades) principales:

1. Tabla de vértices, que almacena la información de los vértices de los puntos de tránsito
2. Tabla de arcos, que almacena los arcos factibles construidos a partir de la tabla de vértices y sus costos
3. Tabla de vértices-cliente, que contiene todos los vértices que corresponden a clientes

A partir de estas se estructuraron las tablas (entidades) complementarias:

1. Ejemplos de prueba, que contiene la información de los ejemplos de prueba como clientes, ventanas de tiempo y número de técnicos.
2. Arcos del grafo completo, que almacena los arcos del grafo completo obtenido sobre cada ejemplo
3. Solución de los ejemplos, que almacena las soluciones de cada uno de los ejemplos antes mencionados

5.3.4 Diseño de la base de datos

Diagrama Entidad-Relación

Ver Figura 5.5, (p.41)

El modelo de entidad relación así como la generación de la base de datos de Access [15] se la realizó en PowerDesigner 9.0.

Tablas de la base de datos

La tabla 5.1 muestra las principales entidades (tablas) de la base de datos de Access.

5.4 Construcción de la red auxiliar

La red inicial se construye tomando los datos de la tablas de vértices, arcos y clientes (sólo los que requieren servicio ese día). Estos datos se exportan a archivos de texto.

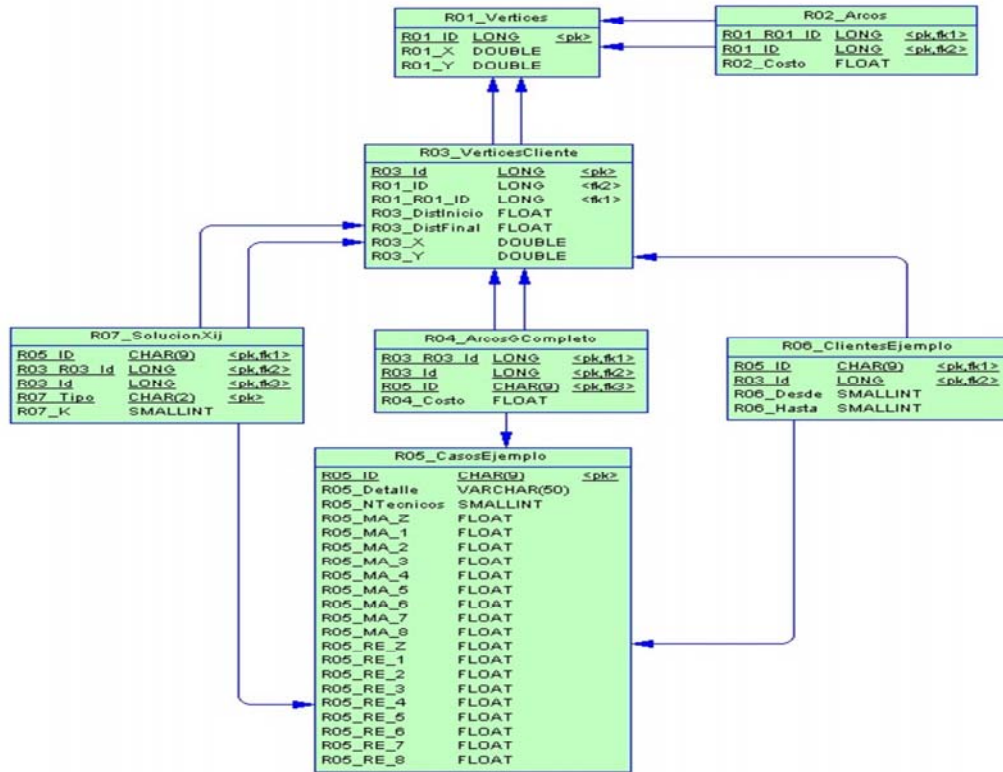


Figura 5.5: Diagrama entidad relación de la base de datos

Nombre de la tabla	Descripción	Referencia
R01_Vertices	Tabla de Vértices (puntos de tránsito)	Tabla 5.2
R02_Arcos	Tabla de Arcos	Tabla 5.3
R03_VerticesCliente	Vértices de los clientes	Tabla 5.4
R04_ArcosGCompleto	Arcos del Grafo Completo	Tabla 5.5
R05_CasosEjemplo	Tabla de casos de ejemplo para pruebas	Tabla 5.6
R06_ClientesEjemplo	Clientes de cada caso de ejemplo	Tabla 5.7
R07_SoluciónXij	Soluciones de los casos de ejemplo	Tabla 5.8

Tabla 5.1: Tablas de la base de datos

Nombre del campo	Descripción
R01_ID	Código del vértice
R01_X	Coordenada Este-Oeste (UTM)
R01_Y	Coordenada Norte-Sur (UTM)

Tabla 5.2: Lista de campos de la tabla R01_Vertices

Nombre del campo	Descripción
R01_ID	Código del vértice de inicio del arco
R01_R01_ID	Código del vértice de llegada del arco
R02_Costo	Costo del arco (metros)

Tabla 5.3: Lista de campos de la tabla R02_Arcos

Nombre del campo	Descripción
R03_ID	Código del cliente - oficina principal
R01_ID	Código del vértice de salida
R01_R01_ID	Código del vértice de llegada
R03_DistInicio	Distancia al vértice de inicio de la red de Quito
R03_DistFinal	Distancia del vértice final de la red de Quito al cliente
R03_X	Coordenada Este - Oeste (UTM)
R03_Y	Coordenada Norte - Sur (UTM)

Tabla 5.4: Lista de campos de la tabla R03_VerticesCliente

Nombre del campo	Descripción
R03_Id	Código del cliente (vértice de salida)
R03_R03_Id	Código del cliente (vértice de llegada)
R05_Id	Código del ejemplo
R04_Costo	Distancia total en metros entre cada uno de los clientes que requieren servicio ese día

Tabla 5.5: Lista de campos de la tabla R04_ArcosGCompleto

Nombre del campo	Descripción
R05_Id	Código del ejemplo
R05_Detalle	Características del ejemplo
R05_NTecnicos	Número de técnicos disponibles
R05_MA_Z	Tiempo total de recorrido (Mod.Mat.)
R05_MA_1	Tiempo de recorrido del técnico 1 (MPE)
R05_MA_2	Tiempo de recorrido del técnico 2 (MPE)
R05_MA_3	Tiempo de recorrido del técnico 3 (MPE)
R05_MA_4	Tiempo de recorrido del técnico 4 (MPE)
R05_MA_5	Tiempo de recorrido del técnico 5 (MPE)
R05_MA_6	Tiempo de recorrido del técnico 6 (MPE)
R05_MA_7	Tiempo de recorrido del técnico 7 (MPE)
R05_MA_8	Tiempo de recorrido del técnico 8 (MPE)
R05_RE_Z	Tiempo total de recorrido (MVTS)
R05_RE_1	Tiempo de recorrido del técnico 1 (MVTS)
R05_RE_2	Tiempo de recorrido del técnico 2 (MVTS)
R05_RE_3	Tiempo de recorrido del técnico 3 (MVTS)
R05_RE_4	Tiempo de recorrido del técnico 4 (MVTS)
R05_RE_5	Tiempo de recorrido del técnico 5 (MVTS)
R05_RE_6	Tiempo de recorrido del técnico 6 (MVTS)
R05_RE_7	Tiempo de recorrido del técnico 7 (MVTS)
R05_RE_8	Tiempo de recorrido del técnico 8 (MVTS)

Tabla 5.6: Lista de campos de la tabla R05_CasosEjemplo

Nombre del campo	Descripción
R05_Id	Código del ejemplo
R03_Id	Código del cliente en el ejemplo
R06_Desde	Límite inferior de la ventana de tiempo
R06_Hasta	Límite superior de la ventana de tiempo

Tabla 5.7: Lista de campos de la tabla R06_ClientesEjemplo

Nombre del campo	Descripción
R05_Id	Código del ejemplo
R03_Id	Código del cliente (vértice de salida)
R03_R03_Id	Código del cliente (vértice de llegada)
R07_Tipo	Tipo de solución (MA o RE)
R07_K	Técnico que viaja del vértice de salida al de llegada

Tabla 5.8: Lista de campos de la tabla R07_SolucionXij

Vértices

Los vértices de la red incluyen:

1. Todos los vértices de la tabla de Vértices (R01_Vertices).
2. Los clientes a ser atendidos ese día (R04_VerticesCliente).
3. Los vértices de la oficina principal 0 y -1

Arcos

Los arcos de la red incluyen:

1. Todos los arcos de la tabla de Arcos (R02_Arcos).
2. Los arcos de entrada y salida del cliente a la RDQ (R04_VerticesCliente).
3. Los arcos de entrada y salida de la oficina a la RDQ (R04_VerticesCliente).

5.5 Construcción de la red principal

La red principal se construye a partir de la red auxiliar considerando lo siguiente:

- Los vértices de la red principal son la oficina (0 y -1) y los clientes de la red auxiliar
- Para cada par de vértices (i, j) que sea compatible, definir un arco en la red principal
- El costo del arco (i, j) es el costo del camino más corto de i a j en la red auxiliar

Para poder cargar esta red en la memoria del computador se utilizaron:

- Las librerías estándar de C++ para el manejo de listas y colas [12].
- Librerías GNU para manejo de grafos y cálculo de distancias mínimas [10].

Los pasos a seguir en el proceso de creación de la red principal se exponen a continuación:

Exportación de los datos iniciales. Se exportó la información de vértices y arcos desde la base de Access a archivos texto con la ayuda de un programa en Visual Basic 6.0.

Se generaron los archivos de texto:

- **Vertices.in**, este archivo contiene la información de todos los vértices de la red auxiliar. Los vértices se listan uno debajo del otro.
- **Arcos.in**, este archivo contiene la información de los arcos de la red auxiliar (vértice inicial, vértice final y costo) separados por “;”

Importación de datos. Se utilizó un programa en C++ para importar las tablas de texto de vértices y arcos (generados en la sección anterior) a una estructura dinámica para el manejo del grafo (listas de adyacencias)

Cálculo de distancias más cortas. El cálculo de las distancias más cortas se hizo mediante el mismo programa anterior en C++ utilizando las librerías estándar de C++ para manejo de listas, colas, grafos y caminos más cortos.

Se utilizó el algoritmo de Dijkstra [4] para caminos más cortos entre todos los clientes y la oficina principal.

Exportación de resultados. Como resultado, el programa antes mencionado exporta el grafo completo entre clientes y la oficina principal a un archivo de texto **Arcos.out**.

Importación de resultados a la base de datos. La tabla resultante del proceso anterior se importa con la ayuda de un programa en Visual Basic 6.0 a la base de Access (Tabla R04_ArcosGCompleto)

5.6 Implementación de los modelos de programación entera

Para la implementación de los modelos de programación entera se utilizó GLPK [13], que es un programa de licencia abierta para la resolución de programas lineales mixtos. Este programa se puede utilizar como una API de C++ o como un archivo ejecutable desde la consola de Windows.

Para el caso de los programas enteros se utilizó el ejecutable.

5.6.1 MPE

El modelo MPE (4.8) se implementó con éxito en ejemplos de pequeña talla (15 clientes y 2 técnicos). Los pasos que se siguieron se detallan a continuación:

1. Crear un archivo de texto con el modelo en formato MathProg, que es uno de los lenguajes que reconoce GLPK. Este archivo de texto se genera con los datos almacenados en la base utilizando un utilitario [14].
2. Ejecutar el solver de GLPK desde la consola: *glpsol -model NombreDelModelo -output ArchivoSalida*
3. Importar el archivo de solución a la base de datos para su análisis utilizando un utilitario [14].

5.6.2 MVTS

El modelo matemático relajado MVTS (4.9) se implementó en ejemplos de pequeña talla (15 clientes y 2 técnicos) y en los ejemplos de estudio (60 clientes y 5-6 técnicos). Los pasos que se siguieron se detallan a continuación:

1. Crear un archivo de texto con el modelo en formato MathProg, que es uno de los lenguajes que reconoce GLPK. Este archivo de texto se genera con los datos almacenados en la base utilizando un utilitario [14].
2. Ejecutar el solver de GLPK desde la consola: *glpsol -model NombreDelModelo -output ArchivoSalida*
3. Importar el archivo de solución a la base de datos para su análisis.
4. Debido a las características de este modelo relajado es necesario crear las rutas basándonos en la solución relajada y mediante el algoritmo de inserción explicado en 4.10 crear un ciclo por técnico. Este algoritmo se implemento en Visual Basic [14].

Capítulo 6

Resultados computacionales

6.1 Origen de los datos

Los datos de los clientes con los que se contruyeron los ejemplos fueron generados aleatoriamente procurando que las ubicaciones geográficas se encuentren dentro del mapa digital de Quito. De igual manera, las ventanas de tiempo se generaron aleatoriamente para el 10% de los clientes en cada ejemplo de prueba.

Se generaron 992 clientes aleatoriamente de los cuales para cada ejemplo se seleccionaron al azar grupos de 15 y 60 clientes y para el 10% de ellos se colocaron límites de tiempo generados aleatoriamente dentro de la jornada de trabajo.

Los ejemplos se identificaron en la base de datos [15] de la siguiente manera: *EJM-NC-NE*. Donde *NC* es el número de clientes que forman parte del ejemplo y *NE* es el número de ejemplo.

6.2 Ejemplos de prueba pequeños (15 clientes y 2 técnicos)

En estos ejemplos se pudo, en la mayoría de casos, emplear con éxito los modelos MPE (4.8) y MVTS (4.9). Los resultados se muestran a continuación:

6.2.1 Comparación de los resultados computacionales de los modelos MPE y MVTS

En la tabla 6.1 se puede apreciar los resultados de ejecutar los modelos MPE y MVTS en ejemplos de talla pequeña. Es claro que el algoritmo MPE se comporta mejor que el modelo relajado MVTS pues los tiempos totales de trabajo de los técnicos son

menores. Esto se puede apreciar en la Figura 6.1, (p.51)

La descripción de cada columna se detalla a continuación:

EJM: Número del ejemplo de prueba para el que se encontró solución

F.O. MPE: Valor de la función objetivo para el modelo MPE (en horas)

F.O. MVTS: Valor de la función objetivo para el modelo MVTS (en horas)

% Error: Porcentaje de error del modelo MVTS con respecto al modelo MPE

Tiempo MPE: Tiempo de ejecución computacional del modelo MPE

Tiempo MVTS: Tiempo de ejecución computacional del modelo MVTS

Ts.MPE: Tiempo sobrante total de los técnicos en el modelo MPE

Ts.MVTS: Tiempo sobrante total de los técnicos en el modelo MVTS

Vt.MVTS: Tiempo de violación de las ventanas de tiempo en el modelo MVTS

En el gráfico comparativo Figura 6.2, (p.52) se pueden apreciar los tiempos sobrantes totales de los técnicos en cada uno de los modelos. En este resultado se puede verificar que el modelo MPE funciona mejor que el MVTS pues los tiempos sobrantes son menores en este último.

6.2.2 Comparación de los itinerarios resultantes de los modelos MPE y MVTS

Para esta comparación se han tomado los dos primeros ejemplos EJM-15-01 y EJM-15-02. Se detallaron los itinerarios en cada caso y se hicieron los análisis que se presentan a continuación.

MPE

Para el ejemplo EJM-15-01 ver la tabla 6.2 y la Figura 6.3, (p.53). Las celdas en amarillos corresponden a clientes que requieren ventanas de tiempo. *TDesplz* es el tiempo que le toma al técnico desplazarse del vértice de inicio al final.

Para el ejemplo EJM-15-02 ver la tabla 6.3 y la Figura 6.4, (p.53). Las celdas en amarillos corresponden a clientes que requieren ventanas de tiempo. *TDesplz* es el tiempo que le toma al técnico desplazarse del vértice de inicio al final.

EJM	F.O. MPE	F.O. MVTS	% de error	Tiempo MPE	Tiempo MVTS	Ts.MPE	Ts.MVTS	Vt.MVTS
1	12.97	13.57	4.6	55	0.0	5.03	4.43	3.94
2	12.56	13.18	4.9	2,456	0.0	5.44	4.82	1.17
3	12.96	14.04	8.3	844	0.3	5.04	3.96	7.37
4	11.78	12.43	5.5	48	0.3	6.22	5.57	4.66
5	12.09	13.15	8.8	947	0.4	5.91	4.85	4.73
7	12.12	12.51	3.2	11	0.3	5.88	5.49	0.00
8	12.13	12.90	6.3	228	0.3	5.87	5.10	3.08
9	12.39	13.49	8.9	639	0.4	5.61	4.51	2.12
10	12.22	13.49	10.4	515	0.4	5.78	4.51	3.69
11	11.72	12.39	5.7	998	0.3	6.28	5.61	2.80
12	12.14	13.37	10.1	833	0.3	5.86	4.63	4.52
13	12.03	13.35	11.0	486	0.3	5.97	4.65	0.71
14	12.53	13.49	7.7	7,546	0.3	5.47	4.51	8.21
16	12.22	13.47	10.2	190	0.3	5.78	4.53	4.70
17	12.28	12.54	2.1	48	0.3	5.72	5.46	0.00
18	11.85	12.73	7.4	1,133	0.3	6.15	5.27	3.57
19	11.99	12.71	6.0	32,451	0.3	6.01	5.29	0.00
21	12.05	12.43	3.2	176	0.3	5.95	5.57	8.72
22	12.63	13.93	10.3	1,200	0.3	5.37	4.55	6.26
23	12.83	13.99	9.0	2,336	0.3	5.17	4.01	5.33
24	11.95	12.40	3.8	349	0.4	6.05	5.60	5.38
27	12.56	13.51	7.6	517	0.3	5.44	4.49	2.68
28	12.37	13.90	12.4	2,711	0.4	5.63	4.10	6.76
29	12.58	13.47	7.1	4,848	0.3	5.42	4.53	3.39
PRM	0.00	0.00	7.3	0	0.0	5.71	4.84	3.91

Tabla 6.1: Comparación de los modelos MPE y MVTS para ejemplos pequeños

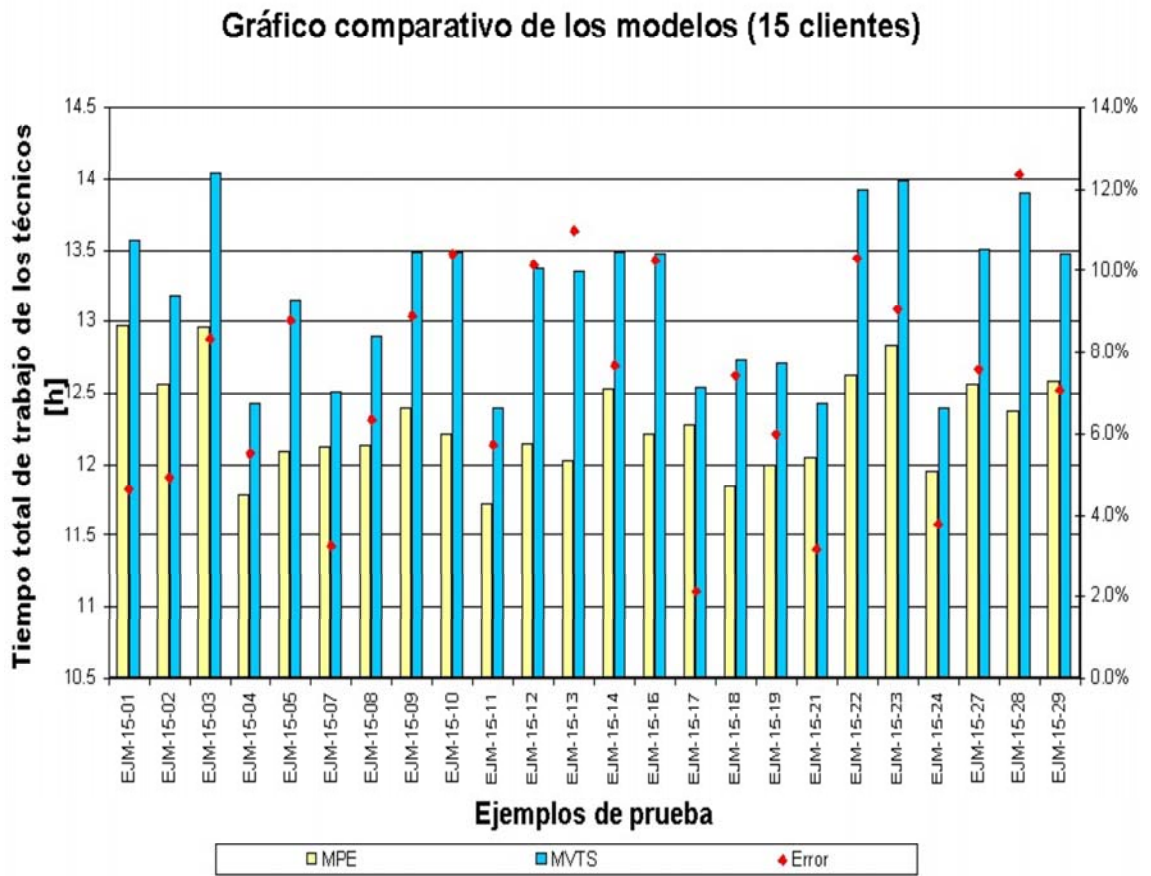


Figura 6.1: Gráfico comparativo entre los modelos MPE y MVTS en tiempos totales (15 clientes)

Inicio	Fin	K	Desde	Hasta	TDesplz	Itinerario
0	-690	1	8	17	0.25	8.75
-690	-562	1	8	17	0.44	9.69
-562	-248	1	8	17	0.42	10.61
-248	-88	1	8	17	0.26	11.37
-88	-325	1	8	17	0.19	12.06
-325	-808	1	12	14	0.22	12.78
-808	-405	1	8	17	0.2	13.48
-405	0	1	8	17	0.2	14.18
0	-647	2	8	17	0.36	8.86
-647	-256	2	8	17	0.13	9.49
-256	-125	2	8	12	0.23	10.22
-125	-707	2	8	17	0.41	11.13
-707	-157	2	8	17	0.34	11.97
-157	-612	2	8	17	0.25	12.72
-612	-511	2	14	18	0.17	14.67
-511	-620	2	8	17	0.13	15.3
-620	0	2	8	17	0.27	16.07

Tabla 6.2: Itinerario del EJM-15-01 en el modelo MPE

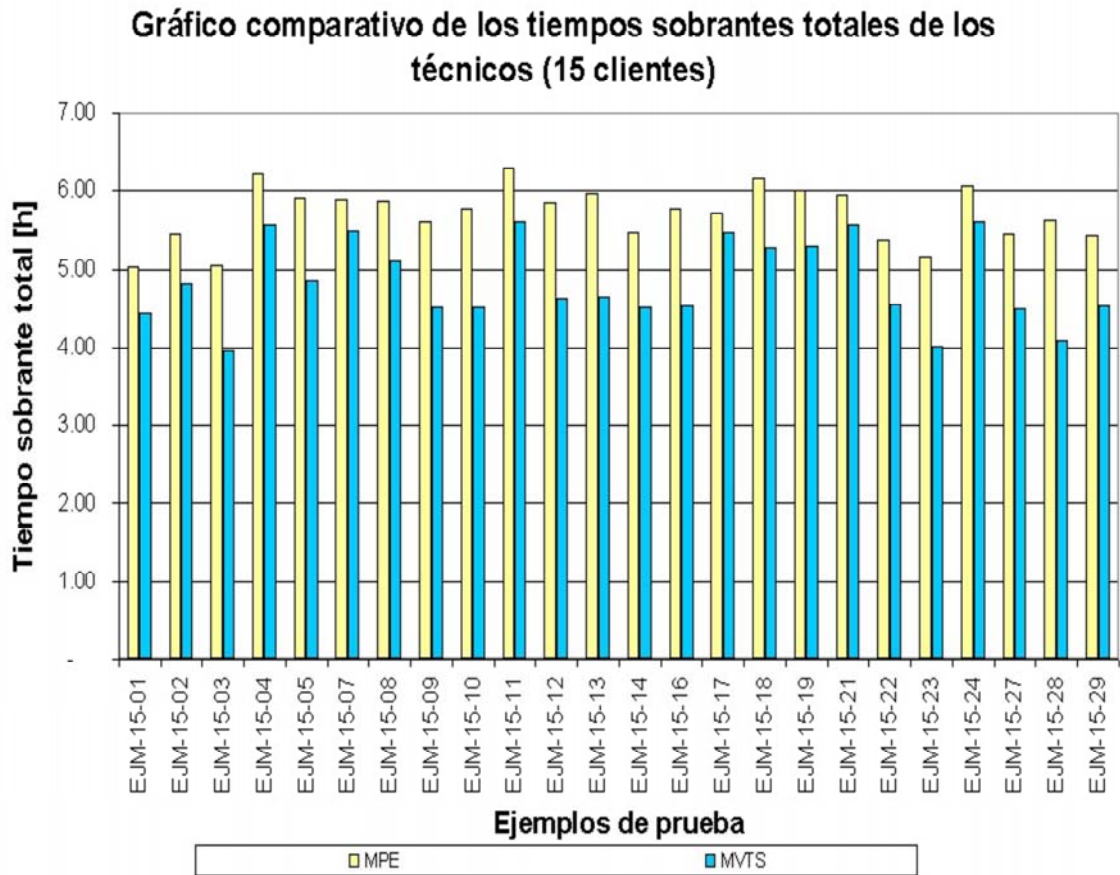


Figura 6.2: Gráfico comparativo entre los modelos en tiempos sobrantes (15 clientes)

Inicio	Fin	K	Desde	Hasta	TDesplz	Itinerario
0	-329	1	8	17	0.11	8.61
-329	-857	1	8	17	0.18	9.29
-857	-889	1	8	17	0.27	10.06
-889	-985	1	8	12	0.27	10.83
-985	-125	1	14	18	0.17	14.67
-125	-854	1	8	17	0.21	15.38
-854	-405	1	8	17	0.07	15.95
-405	0	1	8	17	0.2	16.65
0	-721	2	8	17	0.22	8.72
-721	-642	2	8	17	0.42	9.64
-642	-808	2	12	14	0.13	12.63
-808	-122	2	8	17	0.22	13.35
-122	-88	2	8	17	0.24	14.09
-88	-707	2	8	17	0.53	15.12
-707	-612	2	8	17	0.4	16.02
-612	-838	2	8	17	0.14	16.66
-838	0	2	8	17	0.28	17.44

Tabla 6.3: Itinerario del EJM-15-02 en el modelo MPE

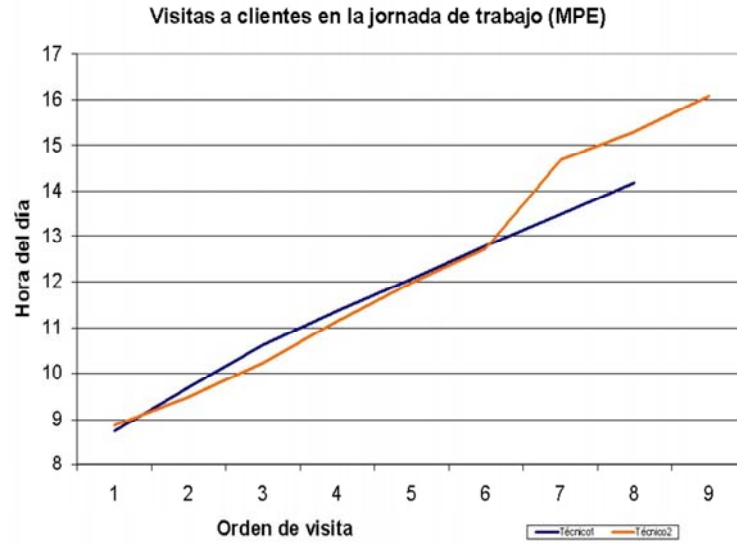


Figura 6.3: Itinerario del EJM-15-01 en el modelo MPE

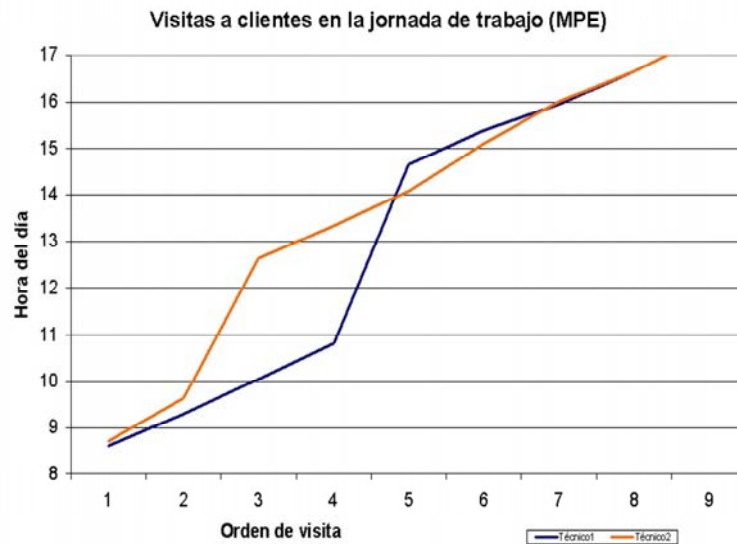


Figura 6.4: Itinerario del EJM-15-02 en el modelo MPE

MVTS

Para el ejemplo EJM-15-01 ver la tabla 6.4 y la Figura 6.5, (p.54). Las celdas en amarillos corresponden a clientes que requieren ventanas de tiempo y las celdas en naranja se presenta cuando se violan ventanas de tiempo. T_{Desplz} es el tiempo que le toma al técnico desplazarse del vértice de inicio al final.

Inicio	Fin	K	Desde	Hasta	TDesplz	Itinerario	Violación
0	-620	1	8	17	0.26	8.76	0
-620	-612	1	8	17	0.13	9.39	0
-612	-511	1	14	18	0.17	10.06	3.94
-511	-157	1	8	17	0.32	10.88	0
-157	-125	1	8	12	0.18	11.56	0
-125	-325	1	8	17	0.28	12.34	0
-325	-248	1	8	17	0.31	13.15	0
-248	-88	1	8	17	0.26	13.91	0
-88	-562	1	8	17	0.52	14.93	0
-562	0	1	8	17	0.7	16.13	0
0	-690	2	8	17	0.25	8.75	0
-690	-707	2	8	17	0.44	9.69	0
-707	-405	2	8	17	0.37	10.56	0
-405	-647	2	8	17	0.17	11.23	0
-647	-256	2	8	17	0.13	11.86	0
-256	-808	2	12	14	0.2	12.56	0
-808	0	2	8	17	0.38	13.44	0

Tabla 6.4: Itinerario del EJM-15-01 en el modelo MVTS

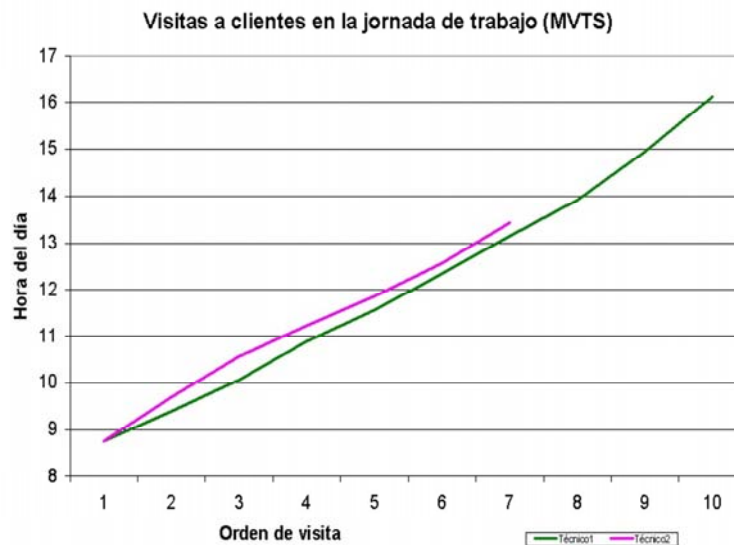


Figura 6.5: Itinerario del EJM-15-01 en el modelo MVTS

Para el ejemplo EJM-15-02 ver la tabla 6.5 y la Figura 6.6, (p.55). Las celdas en amarillos corresponden a clientes que requieren ventanas de tiempo y las celdas en

naranja se presenta cuando se violan ventanas de tiempo. T_{Desplz} es el tiempo que le toma al técnico desplazarse del vértice de inicio al final.

Inicio	Fin	K	Desde	Hasta	TDesplz	Itinerario	Violación
0	-329	1	8	17	0.11	8.61	0
-329	-857	1	8	17	0.18	9.29	0
-857	-612	1	8	17	0.41	10.2	0
-612	-808	1	12	14	0.42	12.92	0
-808	-122	1	8	17	0.22	13.64	0
-122	-88	1	8	17	0.24	14.38	0
-88	-125	1	14	18	0.32	15.2	0
-125	-642	1	8	17	0.25	15.95	0
-642	-707	1	8	17	0.5	16.95	0
-707	0	1	8	17	0.49	17.94	0.94
0	-721	2	8	17	0.22	8.72	0
-721	-854	2	8	17	0.3	9.52	0
-854	-405	2	8	17	0.07	10.09	0
-405	-838	2	8	17	0.28	10.87	0
-838	-889	2	8	17	0.09	11.46	0
-889	-985	2	8	12	0.27	12.23	0.23
-985	0	2	8	17	0.31	13.04	0

Tabla 6.5: Itinerario del EJM-15-02 en el modelo MVTS

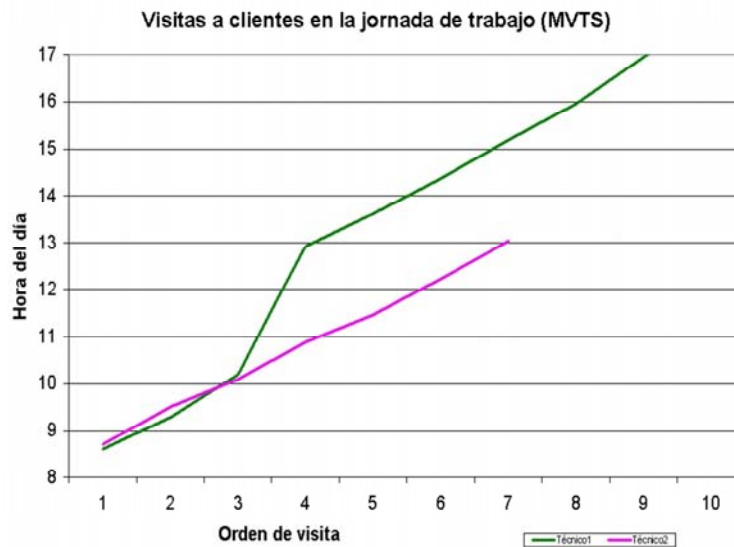


Figura 6.6: Itinerario del EJM-15-02 en el modelo MVTS

6.2.3 Rutas

Las rutas para varios ejemplos se pueden apreciar en el apéndice D.1

6.3 Ejemplos de prueba con 60 clientes y 5-7 técnicos

Para ejemplos de estos tamaños fue imposible correr el modelo MPE pues esto excedía las capacidades del computador. El programa GLPK trataba de encontrar la solución entera hasta saturar por completo la memoria del computador, esto ocurría alrededor de las dos horas de proceso.

6.3.1 Resultados computacionales del modelos MVTS

En la tabla 6.6 se presentan los resultados de cálculo del modelo relajado MVTS conjuntamente con el proceso de inserción que 'arreglaba' las rutas de los técnicos.

La descripción de cada columna se detalla a continuación:

EJM: Número del ejemplo de prueba para el que se encontró solución

Ts.MVTS: Tiempo sobrante total de los técnicos en el modelo MVTS

Tiempo MVTS: Tiempo de ejecución computacional del modelo MVTS

N.Tec: Nmero de técnicos necesarios para servir a los clientes del ejemplo

En el gráfico Figura 6.7, (p.58) se pueden apreciar los técnicos necesarios en cada caso y el tiempo total sobrante.

6.3.2 Itinerario resultante del modelo MVTS

Al igual que en la sección anterior se determinó el itinerario de los técnicos para el primer ejemplo EJM-60-01. Este resultado se puede apreciar en la Figura 6.8, (p.59).

6.3.3 Rutas

Las rutas para varios ejemplos se pueden apreciar en el apéndice D.2

EJM	Ts.MVTS	Tiempo MVTS	N.ºtec.
1	9.38	4.7	7
2	3.30	4.1	7
3	-2.76	4.4	5
4	4.49	4.1	6
5	10.69	4.7	7
6	-0.77	4.1	5
7	7.16	4.6	6
8	-0.40	4.2	5
9	9.99	3.8	7
10	-0.57	3.9	5
11	6.78	4.9	6
12	-1.65	3.8	5
13	0.88	3.9	5
14	4.10	4.3	6
15	-2.77	4.4	5
16	11.87	4.3	7
17	12.85	4.1	7
18	11.19	4.0	7
19	-3.27	4.4	5
20	10.90	4.7	7
21	5.36	3.9	6
22	11.35	3.6	7
23	-0.87	4.8	5
24	4.70	4.5	6
25	-2.61	3.5	5
26	4.66	5.4	6
27	6.48	3.4	6
28	-1.77	4.3	5
29	-2.85	4.4	5
30	10.76	4.9	7
PRM	4.22	4.3	6

Tabla 6.6: Resultados computacionales del modelo MVTS sobre 60 clientes

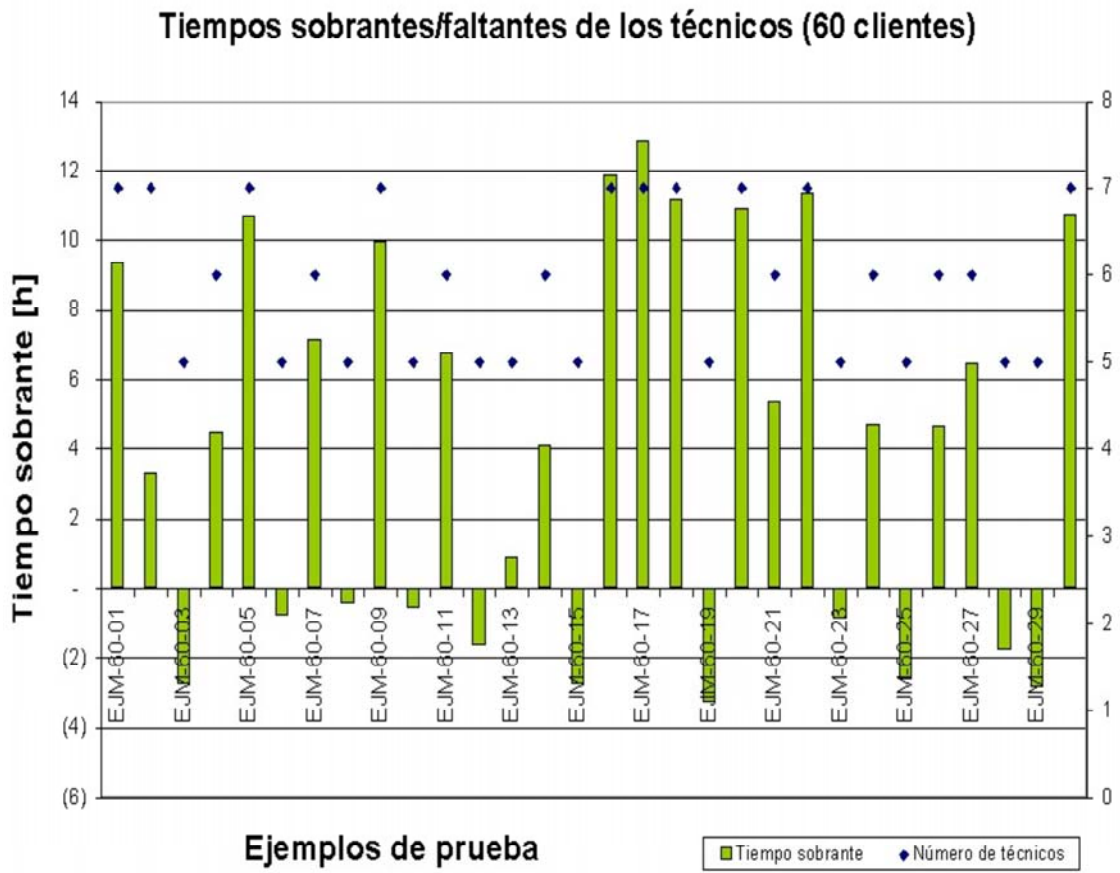


Figura 6.7: Gráfico de técnicos necesarios en cada caso y sus tiempos sobrantes

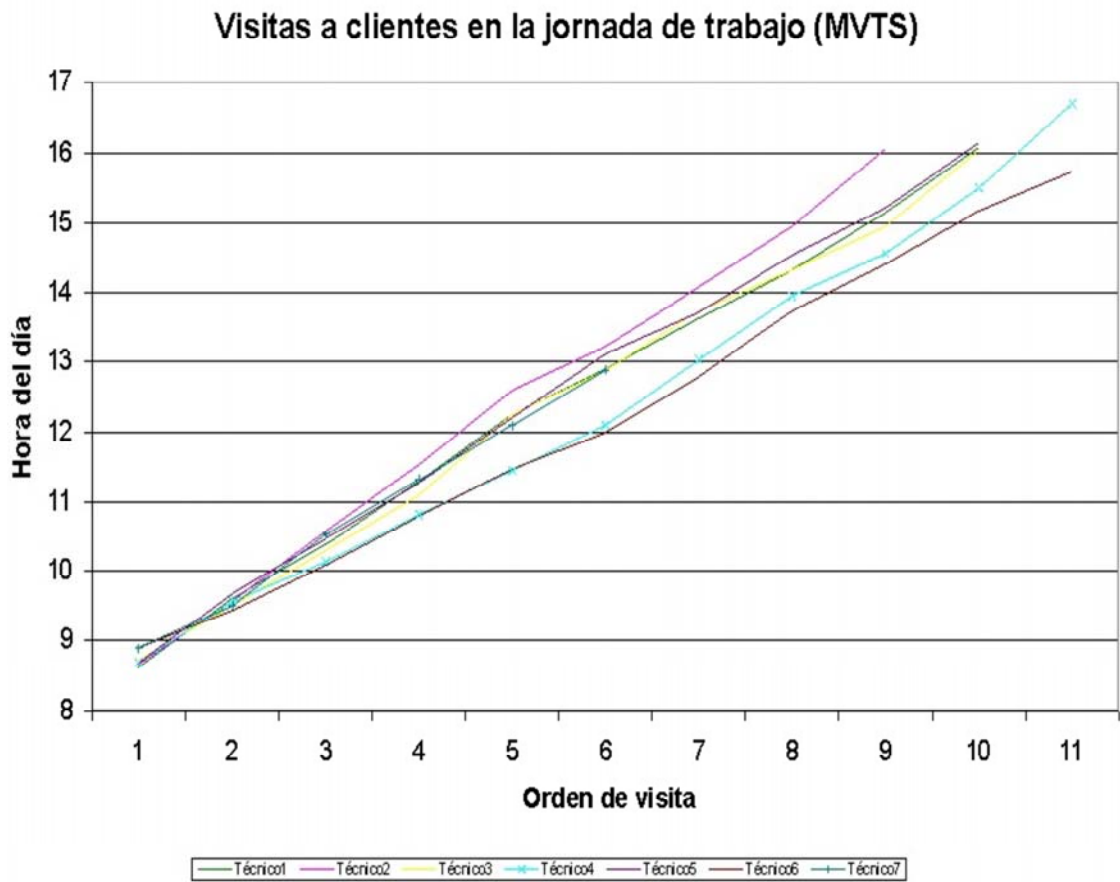


Figura 6.8: Gráfico de técnicos necesarios en cada caso y sus tiempos sobrantes

Capítulo 7

Conclusiones y trabajo futuro

7.1 Conclusiones

1. El modelo de programación entera MPE (4.8) solo puede resolverse de manera directa con el método de 'branch and cut' provisto en GLPK sólo en problemas pequeños.
De los 30 ejemplos de prueba con 15 clientes y 2 técnicos, se encontró solución en 26, en los 4 restantes la potencia de cálculo de la computadora fue insuficiente.
2. En las instancias de mayor talla (60 clientes y 5-7 técnicos) el modelo MEP no puede resolverse por aplicación directa del método de 'branch and cut'. En este caso fue necesario relajar la restricción de ventanas de tiempo e incorporarla con penalizaciones como parte de la función objetivo.
3. En las pruebas hechas sobre los ejemplos de 15 clientes y 2 técnicos se pudo observar que la solución heurística difería en un 7% en promedio con respecto a la solución real variando desde un 2.1% hasta un 12.4%.
4. En la práctica, la solución del PVTs resultó de mayor utilidad, pues esta va asignando el trabajo a cada técnico hasta completar su jornada de trabajo. Hace esto sucesivamente desde el técnico 1 hasta el técnico k , hasta asignar a todos los clientes a un técnico. De esta manera el último técnico, de manera general, dispondrá del mayor tiempo libre, tiempo que será aprovechado por la empresa al asignarle otras tareas de reparación o mantenimiento en la oficina. Es claro que la solución del PVTs viola las ventanas de tiempo, lo cual no es un problema grave pues para la empresa tienen el carácter de recomendable o deseable antes que obligatorio.

5. Es posible resolver este problema directamente aplicando algún algoritmo heurístico de los analizados en 3.4 sin necesidad de resolver un programa entero relajado.

7.2 Trabajo futuro

Tanto en el diseño de los modelos y algoritmos de solución, como en la implementación hay campo para realizar mejoras que podrían ser objeto de futuros trabajos.

Para el caso del diseño de modelos y algoritmos:

- Estudiar la aplicabilidad de modelos basados en rutas, donde las variables de decisión no están asociadas a arcos de la red, sino a rutas factibles para cada técnico. Este tipo de modelos son usados con frecuencia en problemas de enrutamiento, y su solución requiere la aplicación de algún método de generación de columnas.
- Utilizar 'relajación lagrangiana' para mejorar la solución del modelo MVTs.
- Utilizar planos cortantes para encontrar la solución del MPE directamente.
- Utilizar otras heurísticas y metaheurísticas.

En lo concerniente a la implementación

- Integrar todas las aplicaciones en una sola, que se encargue de extraer los datos de la base, calcular la red principal, construir el modelo de optimización y resolverlo empleando el API de GLPK.
- Migrar la aplicación y la base de datos a LINUX, pues es el entorno natural para las herramientas de código abierto. En este caso se propone emplear MySQL como base de datos, desarrollar la aplicación en C/C++ empleando el compilador GCC y acceder directamente al API de GLPK.

Anexo A

Datos

Los datos base para el corrido del programa están almacenados en las tablas R01_Vertices, R02_Arcos y R03_VerticesCliente [15]. A continuación se presenta un extracto de cada una de estas tablas como referencia.

A.1 Tabla de vértices

R01_ID	R01_X	R01_Y	R01_ID	R01_X	R01_Y	R01_ID	R01_X	R01_Y
1	779829.9913	9982232.133	537	778634.0813	9978764.273	1056	779134.2913	9977470.723
4	779867.5913	9982261.253	538	778613.6413	9978757.503	1057	779113.0513	9977453.603
5	779882.9813	9982254.333	539	778653.3213	9978798.503	1058	778262.5713	9977167.833
6	779875.3613	9982231.963	540	778666.5513	9978823.183	1059	778258.1613	9977139.573
7	779858.9913	9982207.983	541	778670.5513	9978799.693	1060	778278.9913	9977098.963
10	779825.4513	9982187.073	542	778675.7713	9978806.863	1061	778250.5613	9977227.953
11	779818.6513	9982192.063	543	778641.7013	9978813.633	1062	778302.2613	9977221.173
12	779820.9213	9982214.423	544	778645.7113	9978820.803	1063	778130.9113	9977448.123
13	780033.3413	9982358.433	545	778693.8113	9978871.753	1064	778145.7413	9977455.283
14	780036.2613	9982370.663	546	778710.2413	9978874.533	1065	778178.2013	9977442.543
15	780049.2213	9982379.513	547	778717.0613	9978882.103	1066	778129.3013	9977415.873
16	780064.2813	9982377.263	548	778709.0413	9978899.613	1067	778149.3313	9977403.533
17	780073.1913	9982365.353	549	778684.1913	9978889.273	1068	778169.7813	9977406.713
18	780067.6813	9982352.313	550	778689.8113	9978896.833	1069	778215.4513	9977340.223
19	780050.6713	9982345.073	551	778625.8913	9977958.603	1070	778146.1213	9977367.703
20	780038.6713	9982344.403	552	778435.2313	9978067.533	1071	778189.4113	9977364.513
21	779849.3313	9982082.953	553	778430.8313	9978083.053	1072	778114.4613	9977377.663
22	779850.4013	9982071.533	554	778421.6013	9978061.563	1073	778110.4613	9977393.583
23	780098.3513	9982037.533	555	778397.5913	9978006.123	1074	778157.7513	9977384.823
24	780095.6813	9982027.713	556	778410.4113	9977981.433	1075	778175.7813	9977347.393
25	780108.5113	9982046.823	557	778385.1613	9977976.663	1076	778091.2213	9977386.823
26	780124.0113	9982045.223	558	778379.5613	9977997.363	1077	778100.0313	9977368.503

Tabla A.1: Resumen de la tabla R01_Vertices

A.2 Tabla de arcos

R01_ID	R01_R01_ID	R02_COSTO	R01_ID	R01_R01_ID	R02_COSTO	R01_ID	R01_R01_ID	R02_COSTO
1	10	45.29	460	461	22.68	998	1001	35.17
1	12	19.89	461	346	68.12	999	1008	85.92
5	4	16.88	462	348	176.11	1000	1001	23.33
5	10	88.5	463	462	27.6	1000	1002	35.6
5	12	73.78	463	464	18.4	1001	792	287.04
6	20	198.28	464	467	60.48	1002	1011	95.38
7	4	53.96	465	464	22.91	1003	999	31.42
7	6	29.03	465	466	25.26	1004	1007	41.11
10	109	256.96	466	547	56.97	1004	1009	21.3
11	4	84.75	467	469	22.05	1005	1009	42.06
11	6	69.34	467	470	21.63	1006	864	107.88
11	10	8.44	467	472	27.02	1007	1038	61.8
12	113	196.13	468	470	27.94	1008	1006	28.66
13	5	182.88	468	472	17.12	1008	1007	18.84
14	13	12.57	469	542	52.47	1009	1000	79.49
14	17	37.32	471	469	33.12	1010	1013	37.95
14	19	29.37	471	472	22.69	1010	1015	22.12
16	13	36.22	472	474	40.31	1011	1015	38.51
16	15	15.23	473	475	28.92	1012	1003	94.8
17	123	29.57	473	476	16.94	1013	1049	76.82
18	15	32.88	474	475	21.04	1014	1012	35.17
18	17	14.15	474	476	25.7	1014	1013	2
19	25	303.81	475	477	189.21	1015	1018	71.88
20	17	40.38	476	480	52.65	1016	1017	2
20	19	12.02	477	483	89.26	1016	1019	34.38
21	7	125.41	479	483	195.79	1017	1014	78.54
22	24	249.16	480	479	21.68	1018	1019	19.16
23	21	253.13	480	482	28.35	1018	1020	35.07
24	28	38.8	481	479	29.23	1019	804	262.1
24	30	10.36	481	482	20.58	1020	1023	56.81
25	23	13.76	482	484	87.07	1021	1017	30.25
25	28	37.74	483	488	79.11	1022	1025	38.78
25	30	27.13	484	487	23.72	1022	1027	22.03
26	18	312.21	485	527	71.74	1023	1027	33.18
27	23	37.9	486	485	26.73	1024	1021	59.12
27	26	19.92	486	487	19	1025	1028	118.43
28	154	475.65	487	489	87.44	1026	1024	31.4
29	26	30.51	488	486	199.76	1026	1025	19.13
29	28	15.3	488	1292	49.73	1027	426	110.58
30	31	223.27	489	491	23.55	1028	1029	20.14
31	36	61.41	490	491	14.62	1028	1031	32.86
31	38	25.09	491	492	73.5	1029	1051	28.8
32	29	220.86	492	493	16.67	1030	1029	25.36
33	32	30.07	492	494	22.64	1031	1032	62.95
33	38	49.89	493	519	67.89	1032	1033	17.95
34	162	472.52	494	498	77.65	1032	1034	41.66
35	32	50.68	495	493	23.13	1033	581	61.15
35	34	22.64	495	494	16.62	1034	718	115.26
36	120	221.78	496	1297	65.03	1035	1033	34.75
37	34	49.17	497	499	22.83	1035	1034	15.85

Tabla A.2: Resumen de la tabla R02_Arcos

A.3 Tabla de Vértices - Cliente

R03_ID	R01_ID	R01_R01_ID	R03_DistInicio	R03_DistFinal	R03_X	R03_Y
0	685	685	42.24	42.24	778347.93	9977383.11
-1	685	685	42.24	42.24	778347.93	9977383.11
-2	403	403	641.41	641.41	780820.65	9977857.43
-3	1147	1146	849.84	850.09	777798.23	9979332.43
-4	305	305	114.16	114.16	779695.64	9980380.1
-5	628	629	842.55	841.24	778143.63	9980782.08
-6	837	837	109.01	109.01	779291.51	9976826.38
-7	354	354	95.23	95.23	780446.49	9978987.39
-8	1468	1468	25.67	25.67	778827.81	9979578.98
-9	606	604	179.6	189.71	779476.02	9981440.38
-10	386	386	60.29	60.29	780020.01	9978040.64
-11	355	355	332.14	332.14	780629.41	9978795.91
-12	355	355	38.9	38.9	780357.38	9978674.37
-13	1381	1381	35.11	35.11	777897.84	9976472.48
-14	454	454	89.54	89.54	778941.92	9978969.72
-15	973	973	77.14	77.14	779542.54	9976475.54
-16	849	849	90.54	90.54	779408.76	9976813.62
-17	319	319	94.93	94.93	779539.85	9979660.65
-18	355	355	174.86	174.86	780471.84	9978762.93
-19	417	417	68.55	68.55	779714.68	9977870.5
-20	116	115	144.31	138.98	779489.75	9982053.84
-21	983	982	727.8	718.32	780214.25	9976302.15
-22	114	115	597.29	585.9	779072.4	9982297.06
-23	320	320	83.98	83.98	779684.11	9979762.46
-24	307	307	13.36	13.36	779541.07	9980181.4
-25	1376	1376	120.73	120.73	777648.62	9976668.33
-26	1147	1146	995.04	996.52	777725.02	9979498.63
-27	983	982	377.08	367.47	780025.91	9976599.24
-28	170	170	141.73	141.73	780584.07	9981512.36
-29	190	190	154.31	154.31	780113.12	9979802.26
-30	194	194	49.66	49.66	780651.35	9980404.66
-31	1488	1488	310.72	310.72	781416.91	9981764.16
-32	116	115	1802.47	1788.93	777834.49	9982154.65
-33	606	607	372.33	384.58	779294.65	9981548.4
-34	353	353	47.52	47.52	780308.53	9978822.15
-35	599	599	78.49	78.49	779603.25	9978165.67
-36	1528	1528	210.48	210.48	779992.48	9978780.28
-37	979	978	212.26	234.42	779075.48	9976184.77
-38	1237	1237	193.87	193.87	777964.5	9977907.06
-39	848	848	57.7	57.7	779022.53	9976719.05
-40	1508	1508	197.98	197.98	781254.19	9980146.83
-41	667	667	31.16	31.16	778400.19	9977436.67
-42	272	272	183.73	183.73	778987.13	9980847.05
-43	628	629	1220.65	1216.41	777680.49	9980443.9
-44	619	618	881.27	883.28	778723.48	9981728.16
-45	1166	1166	54.39	54.39	778663.73	9979080.51
-46	334	334	264.27	264.27	778464.35	9979360.02
-47	628	629	1210.58	1206.18	777690.15	9980411.84
-48	74	74	76.24	76.24	778740.02	9977965.84
-49	431	431	1618.35	1618.35	781409.71	9976461.21

Tabla A.3: Resumen de la tabla R03-VerticesCliente

Anexo B

Puntos de tránsito y movimientos factibles

B.1 Puntos de tránsito

Los puntos de tránsito constituyen la mayoría de los vértices de la red auxiliar y corresponden a sitios precisos identificados sobre el mapa digital de la ciudad de Quito para modelizar los cruces de avenidas y calles. Existen dos tipos de cruces:

- Cruces sin restricción de movimiento. Es decir, los vehículos que llegan a esa intersección pueden tomar cualquiera de las direcciones posibles que salgan de ese punto. (Mientras se respete el sentido de las vías de acceso).
- Cruces con restricciones. En estos lugares los vehículos tienen restricciones de giro a la derecha, giro a la izquierda, giro en 'U' o de seguir en línea recta.

B.1.1 Cruces sin restricciones

Los cruces sin restricciones se modelarán graficando un solo punto de tránsito en el lugar de confluencia de las calles.

Redondel



El redondel es la estructura más simple en la construcción de los puntos de tránsito pues en este punto no se tienen restricciones de giro, es decir el vehículo que llegue a este punto puede salir de él por cualquiera de los arcos de salida. Es claro que en la realidad puede tomar mucho tiempo dar un giro alrededor de un redondel como es el caso de la "Y", pero para efectos de este modelo no se tomará en cuenta esta particularidad.

Cruce de calles secundarias



El cruce de calles secundarias es el más común de los cruces, se da por lo general dentro de sectores residenciales o de poca circulación vehicular.

B.1.2 Cruces con restricciones

En este caso se deben ubicar puntos de tránsito en cada una de las entradas y salidas de las calles o avenidas que se cruzan.

Cruce de avenida principal con calle secundaria



Este tipo de cruces se presenta en todas las intersecciones que se dan a lo largo de una avenida principal. Se da por lo general debido a restricciones de giro de los vehículos que circulan por la avenida hacia las calles secundarias.

Cruce de avenidas principales



Se da en lugares donde confluyen dos avenidas principales.

Cruces complejos



Los cruces complejos se presentan en lugares donde hay más de dos cruces a la vez de avenidas y/o calles secundarias.

B.2 Movimientos factibles

Los arcos de la red representan todos los movimientos factibles que un vehículo puede realizar entre los puntos de tránsito. En particular se deben respetar las siguientes condiciones:

1. Sentido de calles
2. Barreras arquitectónicas y accidentes geográficos
3. Señales de tránsito

B.2.1 Sentido de las calles

Los movimientos factibles entre puntos de tránsito deben respetar el sentido de las calles.

Ejemplo B.1. El presente ejemplo presenta el movimiento prohibido entre dos puntos, dicho movimiento no respeta el sentido de la calle (Sur-Norte). Figura B.1, (p.67). Al construir la red auxiliar, el arco correspondiente no será incluido.



Figura B.1: Sentido de Calles

B.2.2 Barreras arquitectónicas y accidentes geográficos

Los movimientos entre puntos de tránsito deben respetar barreras arquitectónicas y accidentes geográficos (quebradas, ríos, montañas, etc.). Para que un movimiento de este tipo sea factible es indispensable que exista una calle o avenida que conecte de manera directa los dos puntos.

Ejemplo B.2. El presente ejemplo presenta el movimiento no factible entre dos puntos separados por el aeropuerto. El camino alternativo representa la serie de movimientos que deben realizarse en la práctica para rodear el aeropuerto. Figura B.2, (p.68)



Figura B.2: Barreras arquitectónicas

B.2.3 Señales prohibitivas de circulación

En un punto de tránsito pueden existir restricciones de circulación dadas por señales prohibitivas de tránsito. En este caso solo se construirán arcos entre los puntos de cruce cuyos movimientos son factibles.

Ejemplo B.3. Este ejemplo presenta la situación de un vehículo ubicado en la intersección de la avenida 10 de Agosto y la avenida Cristobal Colón. El vehículo se encuentra circulando en dirección Norte-Sur.

Los movimientos permitidos están en azul y los no permitidos en rojo. Figura B.3, (p.69)

La modelización de las alternativas para el vehículo en este cruce se puede apreciar en la Figura B.4, (p.69).

La Figura B.5, (p.69) contiene todos los arcos asociados a este cruce.

Ejemplo B.4. Este ejemplo presenta el cruce de las avenidas Orellana, Eloy Alfaro y 10 de Agosto con todos sus puntos de tránsito y movimientos factibles. Figura B.6, (p.70)

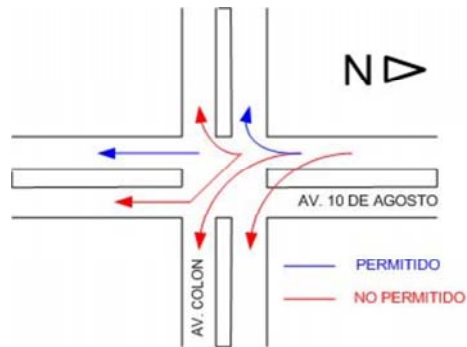


Figura B.3: Situación real



Figura B.4: Movimientos factibles N-S



Figura B.5: Movimientos factibles



Figura B.6: Movimientos en un cruce complejo

Anexo C

Modelo GLPK

C.1 Modelo MPE

C.1.1 Modelo en GLPK

El modelo en GLPK para el MPE de 4.8 (ejemplo 1 con 15 clientes) se lista a continuación:

```
# MODELOS DE RUTAS
#
# Este modelo resuelve al algoritmo de ruteo con ventanas de tiempo para la asignación
# de técnicos a una empresa de seguridad
#
# Referencias:
# Jeysson Vinuesa M.
# Escuela Politécnica Nacional, 2006
#
# glpsol --model Ejm-15-01_MA.mod --output Ejm-15-01_MA.out
/* ----- DEFINICIONES ----- */
set I;
/* clientes factibles y oficina(s) */
set I1;
/* clientes sin vértice -1 */
set I01;
/* clientes sin vértices 0 y -1 */
set K;
/* conjunto de técnicos */

param ti in I, j in I;
```

```

/* tiempo en minutos */

param ai in I;

/* límite inferior de la ventana de tiempo del nodo i */

param bi in I;

/* límite superior de la ventana de tiempo del nodo i */

param Mi in I, j in I;

/* Valor máximo de tiempos disponibles para cada (i,j)*/

var xi in I, j in I, k in K binary;

/* variable binaria para ver si el técnico k, visita el arco (i,j) */

var Ti in I, k in K;

/* Tiempo de arribo del técnico k al nodo i */

/* ----- MODELO ----- */

minimize time: (sumi in I, j in I, k in K t[i,j] * x[i,j,k]);

/* tiempo total del viaje en minutos */

s.t. visitai in I01 : sumj in I, k in K x[i,j,k] = 1;

/* De cada cliente se sale exactamente una sola vez */

s.t. oficinalk in K: sumj in I01 x[0,j,k] = 1;

/* De la oficina sale el técnico una sola vez */

s.t. oficina2k in K: sumi in I01 x[i,-1,k] = 1;

/* A la oficina entra el técnico una sola vez */

s.t. flujoli in I, k in K: sumj in I x[i,j,k] - sumj in I x[j,i,k] = 0;

/* Si el técnico entra a un vértice, debe salir del él */

s.t. ventanali in I1, j in I, k in K: x[i,j,k]*M[i,j] + T[i,k] + 0.5 * t[i,j] - T[j,k] <= M[i,j];

/* Se deben respetar las ventanas de tiempo de todos los clientes */

s.t. ventana2i in I, k in K : a[i] <= T[i,k] <= b[i];

/* El tiempo de arribo de cada técnico debe respetar la ventana de tiempo */

s.t. origenesk in K: x[-1,0,k]=1;

/* Los dos orígenes deben permanecer juntos */

data;

set I := 0 -1 -88 -125 -157 -248 -256 -325 -405 -511 -562 -612 -620 -647 -690 -707 -808 ;

set I1 := 0 -88 -125 -157 -248 -256 -325 -405 -511 -562 -612 -620 -647 -690 -707 -808 ;

set I01 := -88 -125 -157 -248 -256 -325 -405 -511 -562 -612 -620 -647 -690 -707 -808 ;

set K := 1 2 ;

```

```

param a :=
0      8
-1     8
-88    8
-125   8
-157   8
-248   8
-256   8
-325   8
-405   8
-511   14
-562   8
-612   8
-620   8
-647   8
-690   8
-707   8
-808   12;
    
```

```

param b :=
0      17
-1     17
-88    17
-125   12
-157   17
-248   17
-256   17
-325   17
-405   17
-511   18
-562   17
-612   17
-620   17
-647   17
-690   17
-707   17
-808   14;
    
```

```

param t:
0      0      -1      -88      -125      -157      -248      -256      -325      -405      -511      -562      -612      -620      -647      -690      -707      -808 :=
0      10.0    10.0    0.74    0.51    0.41    0.67    0.41    0.61    0.33    0.40    0.78    0.38    0.26    0.36    0.25    0.87    0.54
-1     10.0    10.0    0.74    0.51    0.41    0.67    0.41    0.61    0.33    0.40    0.78    0.38    0.26    0.36    0.25    0.87    0.54
-88    0.68    0.68    10.0    0.32    0.41    0.41    0.56    0.19    0.51    0.63    0.52    0.61    0.63    0.51    0.74    0.53    0.32
-125   0.43    0.43    0.31    10.0    0.17    0.38    0.34    0.28    0.28    0.37    0.49    0.35    0.36    0.27    0.48    0.41    0.25
-157   0.32    0.32    0.41    0.18    10.0    0.43    0.26    0.34    0.20    0.27    0.55    0.25    0.27    0.16    0.39    0.54    0.31
-248   0.59    0.59    0.26    0.36    0.46    10.0    0.48    0.24    0.41    0.61    0.42    0.59    0.61    0.47    0.72    0.59    0.22
-256   0.36    0.36    0.40    0.23    0.30    0.33    10.0    0.27    0.16    0.36    0.44    0.35    0.36    0.23    0.48    0.59    0.20
-325   0.59    0.59    0.28    0.36    0.46    0.31    0.48    10.0    0.41    0.61    0.42    0.59    0.61    0.47    0.72    0.59    0.22
-405   0.20    0.20    0.61    0.35    0.26    0.54    0.27    0.46    10.0    0.32    0.66    0.30    0.28    0.17    0.34    0.71    0.37
-511   0.31    0.31    0.68    0.42    0.32    0.63    0.37    0.55    0.27    10.0    0.75    0.25    0.13    0.26    0.26    0.78    0.46
-562   0.70    0.70    0.37    0.48    0.58    0.42    0.60    0.36    0.52    0.72    10.0    0.71    0.72    0.59    0.84    0.71    0.33
-612   0.34    0.34    0.65    0.39    0.29    0.60    0.33    0.51    0.26    0.17    0.71    10.0    0.17    0.22    0.30    0.74    0.42
-620   0.27    0.27    0.63    0.36    0.27    0.57    0.30    0.48    0.23    0.15    0.69    0.13    10.0    0.20    0.25    0.72    0.40
-647   0.29    0.29    0.50    0.32    0.29    0.42    0.13    0.36    0.11    0.35    0.54    0.33    0.35    10.0    0.44    0.68    0.29
-690   0.44    0.44    0.44    0.44    0.44    0.44    0.44    0.44    0.44    0.44    0.44    0.44    0.44    0.44    10.0    0.44    0.44
-707   0.49    0.49    0.70    0.44    0.34    0.70    0.43    0.62    0.37    0.42    0.82    0.40    0.42    0.33    0.53    10.0    0.53
-808   0.38    0.38    0.32    0.16    0.26    0.24    0.29    0.16    0.20    0.41    0.36    0.39    0.40    0.27    0.52    0.52    10.0;
    
```

```

param M :
    
```

	0	-1	-88	-125	-157	-248	-256	-325	-405	-511	-562	-612	-620	-647	-690	-707	-808 :=
0	19.6	9.60	10.3	10.1	10.0	10.2	10.0	10.2	9.93	4.00	10.3	9.98	9.86	9.96	9.85	10.4	5.14
-1	9.60	19.6	10.3	10.1	10.0	10.2	10.0	10.2	9.93	4.00	10.3	9.98	9.86	9.96	9.85	10.4	5.14
-88	10.2	10.2	19.6	9.92	10.0	10.0	10.1	9.79	10.1	4.23	10.1	10.2	10.2	10.1	10.3	10.1	5.92
-125	5.03	5.03	4.91	14.6	4.77	4.98	4.94	4.88	4.88	0.00	5.09	4.95	4.96	4.87	5.08	5.01	0.85
-157	9.92	9.92	10.0	9.78	19.6	10.0	9.86	9.94	9.80	3.87	10.1	9.85	9.87	9.76	9.99	10.1	5.91
-248	10.1	10.1	9.86	9.96	10.0	19.6	10.0	9.84	10.0	4.21	10.0	10.1	10.2	10.0	10.3	10.1	5.82
-256	9.96	9.96	10.0	9.83	9.90	9.93	19.6	9.87	9.76	3.96	10.0	9.95	9.96	9.83	10.0	10.1	5.80
-325	10.1	10.1	9.88	9.96	10.0	9.91	10.0	19.6	10.0	4.21	10.0	10.1	10.2	10.0	10.3	10.1	5.82
-405	9.80	9.80	10.2	9.95	9.86	10.1	9.87	10.0	19.6	3.92	10.2	9.90	9.88	9.77	9.94	10.3	5.97
-511	10.9	10.9	11.2	20.6	10.9	11.2	10.9	11.1	10.8	14.6	11.3	10.8	10.7	10.8	10.8	11.3	16.60
-562	10.3	10.3	9.97	10.0	10.1	10.0	10.2	9.96	10.1	4.32	19.6	10.3	10.3	10.1	10.4	10.3	5.93
-612	9.94	9.94	10.2	9.99	9.89	10.2	9.93	10.1	9.86	3.77	10.3	19.6	9.77	9.82	9.90	10.3	6.02
-620	9.87	9.87	10.2	9.96	9.87	10.1	9.90	10.0	9.83	3.75	10.2	9.73	19.6	9.80	9.85	10.3	6.00
-647	9.89	9.89	10.1	9.92	9.89	10.0	9.73	9.96	9.71	3.95	10.1	9.93	9.95	19.6	10.0	10.2	5.89
-690	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	4.04	10.0	10.0	10.0	10.0	19.6	10.0	5.04
-707	10.0	10.0	10.3	10.0	9.94	10.3	10.0	10.2	9.97	4.02	10.4	10.0	10.0	9.93	10.1	19.6	6.13
-808	5.98	5.98	6.92	16.6	6.86	5.84	5.89	6.76	6.80	1.01	5.96	5.99	7.00	6.87	7.12	7.12	12.6;
end;																	

C.1.2 Solución del modelo en GLPK

A continuación se lista una parte de la salida obtenida por GLPK:

No.	Column name	Activity	Lower bound	Upper bound
27	x[0,-647,1]	* 0	0	1
28	x[0,-647,2]	* 1	0	1
29	x[0,-690,1]	* 1	0	1
30	x[0,-690,2]	* 0	0	1
31	x[0,-707,1]	* 0	0	1
32	x[0,-707,2]	* 0	0	1
33	x[0,-808,1]	* 0	0	1
34	x[0,-808,2]	* 0	0	1
35	x[-1,0,1]	* 1	0	1
36	x[-1,0,2]	* 1	0	1
37	x[-1,-1,1]	* 0	0	1
38	x[-1,-1,2]	* 0	0	1

Cada uno de los campos se detalla a continuación:

No. Representa el número de línea del archivo solución

Column name Representa la variable x_{ijk} con sus respectivos índices

Activity Representa el valor que toma la variable x_{ijk} en este caso 0 ó 1

Lower bound Representa el límite inferior de la variable x_{ijk}

Upper bound Representa el límite superior de la variable x_{ijk}

C.2 Modelo MVTs

C.2.1 Modelo en GLPK

El modelo en GLPK para 4.9 se lista a continuación:

```
# MODELOS DE RUTAS

#

# Este modelo resuelve al algoritmo de ruteo con ventanas de tiempo para la asignación
# de técnicos a una empresa de seguridad (Modelo relajado)

#

# Referencias:

# Jeysson Vinuesa M.
# Escuela Politécnica Nacional, 2006

#

# glpsol --model Ejm-15-01_RE.mod --output Ejm-15-01_RE.out

/* ----- DEFINICIONES ----- */

set I;

/* clientes factibles y oficina(s) */

set I1;

/* clientes sin vértice -1 */

set IO1;

/* clientes sin vértices 0 y -1 */

set K;

/* conjunto de técnicos */

param ti in I, j in I;

/* tiempo en minutos */

param ai in I;

/* límite inferior de la ventana de tiempo del nodo i */

param bi in I;

/* límite superior de la ventana de tiempo del nodo i */

param Mi in I, j in I;

/* Valor máximo de tiempos disponibles para cada (i,j)*/

var xi in I, j in I, k in K binary;

/* variable binaria para ver si el técnico k, visita el arco (i,j) */

var Ti in I, k in K;
```

```

/* Tiempo de arribo del técnico k al nodo i */

/* ----- MODELO ----- */

minimize time: (sumi in I, j in I, k in K (t[i,j] * x[i,j,k])) + (sumi in I1, j in I, k in K (x[i,j,k]*M[i,j] + T[i,k] + 0.5 * t[i,j] -
T[j,k] - M[i,j]));

/* tiempo total del viaje en minutos */

s.t. visitai in I01 : sumj in I, k in K x[i,j,k] = 1;

/* De cada cliente se sale exactamente una sola vez */

s.t. oficinalk in K: sumj in I01 x[0,j,k] = 1;

/* De la oficina sale al técnico una sola vez */

s.t. oficina2k in K: sumi in I01 x[i,-1,k] = 1;

/* A la oficina entra el técnico una sola vez */

s.t. flujoli in I, k in K: sumj in I x[i,j,k] - sumj in I x[j,i,k] = 0;

/* Si el técnico entra a un vértice, debe salir del él */

/*s.t. ventanali in I1, j in I, k in K: x[i,j,k]*M[i,j] + T[i,k] + 0.5 * t[i,j] - T[j,k] <= M[i,j];*/

/* Se deben respetar las ventanas de tiempo de todos los clientes */

s.t. ventana2i in I, k in K : a[i] <= T[i,k] <= b[i];

/* El tiempo de arribo de cada técnico debe respetar la ventana de tiempo */

s.t. origenesk in K: x[-1,0,k]=1;

/* Los dos origenes deben permanecer juntos */

data;

set I := 0 -1 -88 -125 -157 -248 -256 -325 -405 -511 -562 -612 -620 -647 -690 -707 -808 ;

set I1 := 0 -88 -125 -157 -248 -256 -325 -405 -511 -562 -612 -620 -647 -690 -707 -808 ;

set I01 := -88 -125 -157 -248 -256 -325 -405 -511 -562 -612 -620 -647 -690 -707 -808 ;

set K := 1 2 ;

param a :=
0      8
-1     8
-88    8
-125   8
-157   8
-248   8
-256   8
-325   8
-405   8
-511   14
-562   8
-612   8
-620   8
-647   8
-690   8
-707   8
-808   12;

```

```
param b :=
0      17
-1     17
-88    17
-125   12
-157   17
-248   17
-256   17
-325   17
-405   17
-511   18
-562   17
-612   17
-620   17
-647   17
-690   17
-707   17
-808   14;
```

```
param t:
0      0      -1      -88      -125      -157      -248      -256      -325      -405      -511      -562      -612      -620      -647      -690      -707      -808 :=
0      10.0   10.0   0.74   0.51   0.41   0.67   0.41   0.61   0.33   0.40   0.78   0.38   0.26   0.36   0.25   0.87   0.54
-1     10.0   10.0   0.74   0.51   0.41   0.67   0.41   0.61   0.33   0.40   0.78   0.38   0.26   0.36   0.25   0.87   0.54
-88    0.68   0.68   10.0   0.32   0.41   0.41   0.56   0.19   0.51   0.63   0.52   0.61   0.63   0.51   0.74   0.53   0.32
-125   0.43   0.43   0.31   10.0   0.17   0.38   0.34   0.28   0.28   0.37   0.49   0.35   0.36   0.27   0.48   0.41   0.25
-157   0.32   0.32   0.41   0.18   10.0   0.43   0.26   0.34   0.20   0.27   0.55   0.25   0.27   0.16   0.39   0.54   0.31
-248   0.59   0.59   0.26   0.36   0.46   10.0   0.48   0.24   0.41   0.61   0.42   0.59   0.61   0.47   0.72   0.59   0.22
-256   0.36   0.36   0.40   0.23   0.30   0.33   10.0   0.27   0.16   0.36   0.44   0.35   0.36   0.23   0.48   0.59   0.20
-325   0.59   0.59   0.28   0.36   0.46   0.31   0.48   10.0   0.41   0.61   0.42   0.59   0.61   0.47   0.72   0.59   0.22
-405   0.20   0.20   0.61   0.35   0.26   0.54   0.27   0.46   10.0   0.32   0.66   0.30   0.28   0.17   0.34   0.71   0.37
-511   0.31   0.31   0.68   0.42   0.32   0.63   0.37   0.55   0.27   10.0   0.75   0.25   0.13   0.26   0.26   0.78   0.46
-562   0.70   0.70   0.37   0.48   0.58   0.42   0.60   0.36   0.52   0.72   10.0   0.71   0.72   0.59   0.84   0.71   0.33
-612   0.34   0.34   0.65   0.39   0.29   0.60   0.33   0.51   0.26   0.17   0.71   10.0   0.17   0.22   0.30   0.74   0.42
-620   0.27   0.27   0.63   0.36   0.27   0.57   0.30   0.48   0.23   0.15   0.69   0.13   10.0   0.20   0.25   0.72   0.40
-647   0.29   0.29   0.50   0.32   0.29   0.42   0.13   0.36   0.11   0.35   0.54   0.33   0.35   10.0   0.44   0.68   0.29
-690   0.44   0.44   0.44   0.44   0.44   0.44   0.44   0.44   0.44   0.44   0.44   0.44   0.44   0.44   10.0   0.44   0.44
-707   0.49   0.49   0.70   0.44   0.34   0.70   0.43   0.62   0.37   0.42   0.82   0.40   0.42   0.33   0.53   10.0   0.53
-808   0.38   0.38   0.32   0.16   0.26   0.24   0.29   0.16   0.20   0.41   0.36   0.39   0.40   0.27   0.52   0.52   10.0;
```

```
param M :
0      0      -1      -88      -125      -157      -248      -256      -325      -405      -511      -562      -612      -620      -647      -690      -707      -808 :=
0      19.6   19.6   10.3   10.1   10.0   10.2   10.0   10.2   9.93   4.00   10.3   9.98   9.86   9.96   9.85   10.4   5.14
-1     19.6   19.6   10.3   10.1   10.0   10.2   10.0   10.2   9.93   4.00   10.3   9.98   9.86   9.96   9.85   10.4   5.14
-88    10.2   10.2   19.6   9.92   10.0   10.0   10.1   9.79   10.1   4.23   10.1   10.2   10.2   10.1   10.3   10.1   5.92
-125   5.03   5.03   4.91   14.6   4.77   4.98   4.94   4.88   4.88   0.00   5.09   4.95   4.96   4.87   5.08   5.01   0.85
-157   9.92   9.92   10.0   9.78   19.6   10.0   9.86   9.94   9.80   3.87   10.1   9.85   9.87   9.76   9.99   10.1   5.91
-248   10.1   10.1   9.86   9.96   10.0   19.6   10.0   9.84   10.0   4.21   10.0   10.1   10.2   10.0   10.3   10.1   5.82
-256   9.96   9.96   10.0   9.83   9.90   9.93   19.6   9.87   9.76   3.96   10.0   9.95   9.96   9.83   10.0   10.1   5.80
-325   10.1   10.1   9.88   9.96   10.0   9.91   10.0   19.6   10.0   4.21   10.0   10.1   10.2   10.0   10.3   10.1   5.82
-405   9.80   9.80   10.2   9.95   9.86   10.1   9.87   10.0   19.6   3.92   10.2   9.90   9.88   9.77   9.94   10.3   5.97
-511   10.9   10.9   11.2   20.6   10.9   11.2   10.9   11.1   10.8   14.6   11.3   10.8   10.7   10.8   10.8   11.3   16.60
-562   10.3   10.3   9.97   10.0   10.1   10.0   10.2   9.96   10.1   4.32   19.6   10.3   10.3   10.1   10.4   10.3   5.93
-612   9.94   9.94   10.2   9.99   9.89   10.2   9.93   10.1   9.86   3.77   10.3   19.6   9.77   9.82   9.90   10.3   6.02
-620   9.87   9.87   10.2   9.96   9.87   10.1   9.90   10.0   9.83   3.75   10.2   9.73   19.6   9.80   9.85   10.3   6.00
-647   9.89   9.89   10.1   9.92   9.89   10.0   9.73   9.96   9.71   3.95   10.1   9.93   9.95   19.6   10.0   10.2   5.89
-690   10.0   10.0   10.0   10.0   10.0   10.0   10.0   10.0   10.0   4.04   10.0   10.0   10.0   10.0   19.6   10.0   5.04
-707   10.0   10.0   10.3   10.0   9.94   10.3   10.0   10.2   9.97   4.02   10.4   10.0   10.0   9.93   10.1   19.6   6.13
-808   5.98   5.98   6.92   16.6   6.86   5.84   5.89   6.76   6.80   1.01   5.96   5.99   7.00   6.87   7.12   7.12   12.6;
```

end;

Anexo D

Mapas Resultantes

D.1 Rutas por cada técnico (15 clientes)

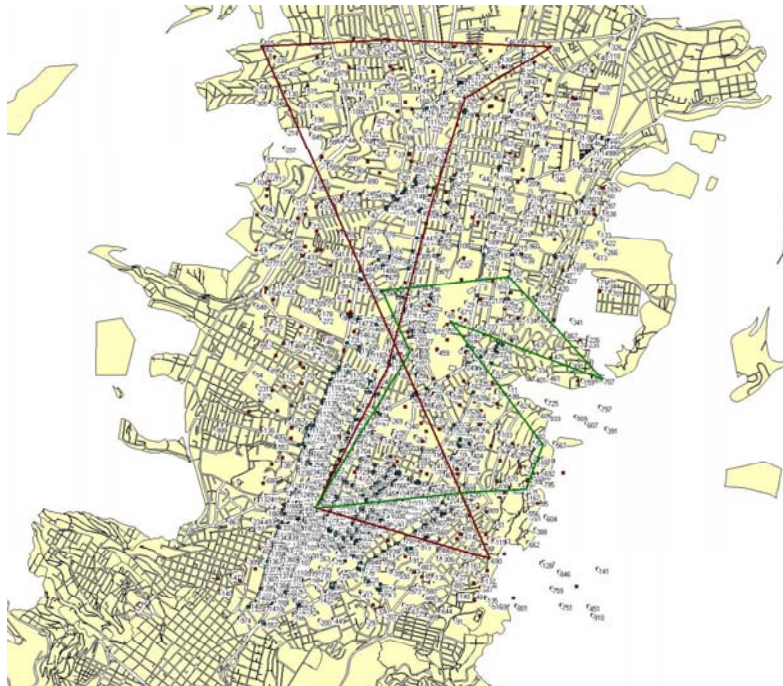


Figura D.1: Mapa de las rutas del ejemplo EJM-15-01 (MPE)

D.2 Rutas por cada técnico (60 clientes)



Figura D.2: Mapa de las rutas del ejemplo EJM-15-01 (MVTs)



Figura D.3: Mapa de las rutas del ejemplo EJM-15-02 (MPE)



Figura D.4: Mapa de las rutas del ejemplo EJM-15-02 (MVTs)



Figura D.5: Mapa de las rutas del ejemplo EJM-15-03 (MPE)



Figura D.6: Mapa de las rutas del ejemplo EJM-15-03 (MVTs)



Figura D.7: Mapa de las rutas del ejemplo EJM-15-04 (MPE)



Figura D.8: Mapa de las rutas del ejemplo EJM-15-04 (MVTs)



Figura D.9: Mapa de las rutas del ejemplo EJM-60-01 (MVTs)



Figura D.10: Mapa de las rutas del ejemplo EJM-60-02 (MVTS)



Figura D.11: Mapa de las rutas del ejemplo EJM-60-03 (MVTS)



Figura D.12: Mapa de las rutas del ejemplo EJM-60-04 (MVTS)

Bibliografía

- [1] Torres, Luis Miguel. *Online Vehicle Routing*. Cuvillier Verlag Göttingen, Berlin, 2004.
- [2] Edmonds, Jack. *Paths, trees and flowers*. O.N.R. Logistics Project at Princeton, 1963.
- [3] Sakarovitch M., *Eléments de la theoria des Graphes*, Institut National Polytechnique de Grenoble, 1977.
- [4] Sakarovitch M., *Optimisation Combinatoire*, Institut National Polytechnique de Grenoble, 1983.
- [5] Ross, Kennet A. & Wright, Charles R.B. *Discrete Mathematics*, Prentice Hall, New Jersey, 1988.
- [6] Molluzzo, John & Buckley, Fred. *A First Course in Discrete Mathematics*, Wadsworth Publishing Company, Belmont, California, 1986
- [7] Ball, M.O. & Magnati, T.L. & Monma, C.L. & Nemhauser, G.L. , *Network Routing*, Elsevier Science B.V., Amsterdam, 1995
- [8] Stroustrup, Bjarne. *The C++ Programming Language*, AT & T Labs, Murray Hill, New Jersey, 3rd edition, 1997
- [9] Eckel, Bruce. *Thinking in C++*, President MindView Inc., 2nd edition, 1999
- [10] Iovine, Francesco. <http://gauguin.info.uniroma2.it/italiano/Teaching/Algoritmi/asd/doc/grafo/Grafo.html>, Sep., 2005
- [11] Toth Paolo & Vigo, Daniele. *The Vehicle Routing Problem*, SiamBooks, Philadelphia, 2002.

- [12] Standard Template Library, [http:// www-306.ibm.com / software / awdtools / c390 / features / cmvsstlp.html](http://www-306.ibm.com/software/awdtools/c390/features/cmvsstlp.html)
- [13] GNU Linear Programming Kit, [http:// www.gnu.org / software / glpk / glpk.html](http://www.gnu.org/software/glpk/glpk.html)
- [14] Rutas.Exe, Interfase en Visual Basic 6.0 para el manejo de la base de datos de Access.
- [15] RedQuito.mdb, Base de datos en Access 2000 donde se almacenan las tablas de vértices, arcos, etc.