

**UNIVERSIDAD DE CONCEPCIÓN - CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL**

**OPTIMIZANDO LA PROGRAMACIÓN DE
AUDITORÍAS DE UNA EMPRESA DE
CERTIFICACIÓN EN ECUADOR**

Por

Luis Javier Aguilar De La Torre

Profesora Guía

Dra. Lorena Pradenas Rojas

Concepción, Abril de 2020

Tesis presentada a la

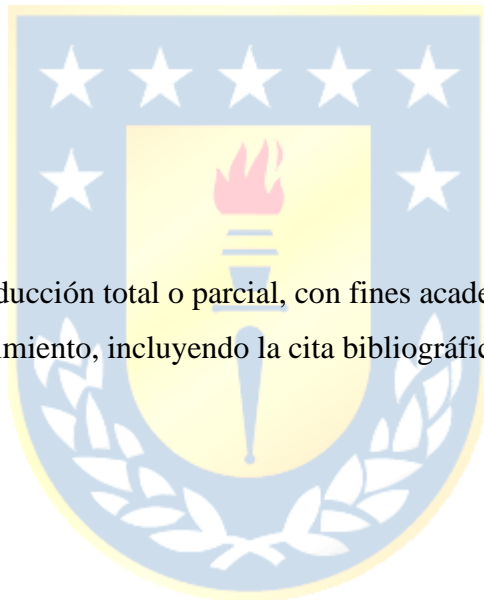
**DIRECCIÓN DE POSTGRADOS
UNIVERSIDAD DE CONCEPCIÓN**



Para optar al grado de

MAGÍSTER EN INGENIERÍA INDUSTRIAL

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento



DEDICATORIA

A mi esposa e hija, el motor de todos mis actos. A la memoria de mi padre, Luis Alberto, y al apoyo constante de mi madre, Rocío. A mi Profesora Guía, gracias por su paciencia, sin su soporte la culminación de este trabajo no habría sido posible.



RESUMEN

OPTIMIZANDO LA PROGRAMACIÓN DE AUDITORÍAS DE UNA EMPRESA DE CERTIFICACIONES EN ECUADOR

Luis Aguilar De la Torre

Abril 2020

Profesor guía: Lorena Pradenas Rojas

Programa: Magíster en Ingeniería Industrial

El presente estudio aborda el problema de programación de auditorías en locales de una franquicia de alimentación, atendidos por una empresa de certificaciones en Ecuador; que es resuelto mediante una variante del ampliamente conocido Problema del Vendedor viajero o TSP, específicamente el MTSP (*Multiple Traveling Salesman Problem*). Con el fin de facilitar su resolución por métodos exactos, se realiza la clusterización de los 48 locales, repartidos en 27 ciudades de Ecuador, por zona geográfica y por ciudad, adoptando un enfoque generalizado de solución.

El modelo matemático desarrollado incorpora restricciones que permiten cumplir con una característica específica del problema, que es la duración de la jornada laboral de los auditores, sujeta a la regulación local. Para ello, se obtiene una solución inicial mediante el uso de una heurística de tipo *Nearest Neighbor* (NN), con la que se estima el número de ciclos de auditoría necesarios para cumplir con esta condición. Este número de ciclos se puede reducir en busca de la solución óptima. Como resultado se obtienen las rutas y ciclos óptimos para los 2 circuitos generados por la clusterización de locales, esta solución mejora significativamente la solución inicial obtenida por métodos heurísticos.

Palabras claves: programación de auditorías, MTSP.

ABSTRACT

OPTIMIZING THE SCHEDULE OF AUDITS OF A CERTIFICATION COMPANY IN ECUADOR

Luis Aguilar De la Torre

Abril 2020

Thesis Supervisor: Lorena Pradenas Rojas

Program: Magíster en Ingeniería Industrial

This study addresses the problem of scheduling audits in premises of a fast food franchise, attended by a certification company in Ecuador, which is solved by a variant of the widely known Traveling Salesman Problem or TSP, specifically the MTSP. In order to facilitate the resolution by exact methods, two circuits are conformed with the 48 selected premises, spread over 27 cities in Ecuador, considering the geographical area and the city, adopting for each one a generalized solution approach.

The developed mathematical model incorporates restrictions that allow compliance with a specific characteristic of the problem, which is the length of the auditor's working week, subject to the local labor regulation. For this, the estimated number of audit cycles that meet this condition per circuit, is obtained through an initial heuristic solution (Nearest Neighbor). The number of cycles obtained with the initial solution can be reduced by the model in the search of the optimal solution. As result, optimal routes and cycles are obtained, for the 2 circuits generated by the clustering of premises, this solution significantly improves the initial solution obtained by heuristic methods

Key words: Audit scheduling, MTSP.

TABLA DE CONTENIDO

DEDICATORIA.....	II
RESUMEN.....	IV
TABLA DE CONTENIDO.....	VI
ÍNDICE DE TABLAS.....	VIII
INDICE DE ILUSTRACIONES.....	IX

CAPÍTULO 1

Introducción.....	10
1.1 Descripción del problema.....	11
1.2 Requerimientos.....	14
1.3 Hipótesis de Investigación.....	15
1.4 Objetivos.....	15
1.4.1 Objetivo general.....	15
1.4.2 Objetivos específicos.....	15
1.5 Plan de acción.....	16

CAPÍTULO 2

Marco teórico.....	19
2.1 Historia del TSP.....	19
2.2 Modelado y formulación del TSP.....	20
2.3 Variantes del TSP.....	22

CAPÍTULO 3

Estado del arte.....	24
3.1 Solucionadores no exactos.....	24
3.2 Solucionadores exactos.....	26

CAPÍTULO 4

Metodología.....	28
4.1 Definición de los circuitos.....	28
4.2 Modelo matemático.....	30
4.2.1 Acotamiento de ciclos.....	33
4.3 Solución inicial.....	34

CAPÍTULO 5

Resultados	36
5.1 Solución inicial	36
5.1.1 Resultados iniciales Circuito Norte.....	36
5.1.1 Resultados iniciales Circuito Sur	37
5.2 Solución del modelo matemático.....	39
5.2.1 Resultados circuito norte.....	39
5.2.1 RESULTADOS CIRCUITO SUR	41

CAPÍTULO 6

Discusión.....	43
----------------	----

CAPÍTULO 7

Conclusiones	48
Bibliografía	50
Anexos	53
Anexo 1. Locales en Ecuador de la franquicia <i>Fast Food Inc.</i> (fuera de Quito y Guayaquil)	53
Anexo 2. Matriz de distancias y duración de auditorías para el Circuito Norte (minutos)	55
Anexo 3. Matriz de distancias y duración de auditorías para el Circuito Sur (minutos)	56
Anexo 4. Modelación en LINGO del problema SD-MTSP desarrollado.	57
Anexo 5. Modelación en LINGO del problema TSP.	59

ÍNDICE DE TABLAS

Tabla 1.1 Ejemplo de una jornada semanal de auditorías	13
Tabla 1.2 Distribución de los locales en estudio	13
Tabla 1.3 Distribución de los locales del Circuito Norte	17
Tabla 1.4 Distribución de los locales del Circuito Sur.....	18
Tabla 3.1 Hitos en el desarrollo del TSP (Chauhan et al., 2012).....	27
Tabla 4.1 Distribución de los locales del Circuito Norte	29
Tabla 4.2 Algoritmo NN utilizado para la solución inicial	35
Tabla 5.1 Solución inicial Circuito Norte	37
Tabla 5.2 Solución inicial Circuito Sur	37
Tabla 5.3 Matriz de solución inicial Circuito Norte	38
Tabla 5.4 Matriz de solución inicial Circuito Sur	38
Tabla 5.5 Solución Circuito Norte	39
Tabla 5.6 Comparación de resultados Circuito Norte	40
Tabla 5.7 Solución Circuito Sur	41
Tabla 5.8 Comparación de resultados Circuito Sur	42
Tabla 6.1 Tiempo computacional por instancia	44
Tabla 6.2 Respuesta del modelo a la variación de número de ciclos	45
Tabla 6.3 Respuesta del modelo a la amplitud de ciclo	46

INDICE DE ILUSTRACIONES

Figura 1.2 Representación de los Circuitos Norte y Sur	17
Figura 2.1 Variantes del TSP: Máximo, cuello de botella y gráfico.....	23
Figura 2.2 Variantes del TSP: Agrupado y Generalizado.....	23
Figura 2.3 Variantes del TSP: Múltiples Viajantes.....	23
Figura 5.1 Representación del Circuito Norte	40
Figura 5.2 Representación del Circuito Sur	42
Figura 6.1 Tiempo computacional por instancia, por método de solución	43



CAPÍTULO 1

Introducción

La actual configuración de la cadena de suministro de alimentos genera múltiples desafíos para aquellos grupos de interés, preocupados en el aseguramiento de su inocuidad, y de otras características relevantes como su calidad, legalidad y autenticidad. Factores como las grandes distancias que recorren los alimentos desde su producción primaria hasta su consumo, largos periodos de conservación de algunos tipos de productos, y el gran número de involucrados en los diferentes eslabones que componen su cadena de valor; generan condiciones desfavorables en la aplicación de controles efectivos para garantizar la seguridad de los consumidores finales (Stringer, 2005).

Como respuesta a estas condiciones y a los cambios constantes en el entorno de la producción, comercialización y consumo de alimentos, se ha desarrollado un variado número de estándares auditables de inocuidad, que además de establecer condiciones mínimas aceptables para: la cosecha, manipulación, procesamiento, transporte y expendio de alimentos; permiten a las industrias de este sector adoptar un enfoque estandarizado de manejo de riesgos (Trienekens & Zuurbier, 2008).

Sin embargo, entre 97 y 128 firmas a nivel mundial proveen servicios de auditorías de inocuidad alimentaria, según los directorios de los principales estándares de certificación en este rubro (*IFS Database, 2020*) (*BRC Directory, 2020*); muchas de éstas en Ecuador, y al igual que otras organizaciones, se enfrentan a presiones competitivas para reducir costos, mejorar la calidad y aumentar la productividad. Una forma en que las empresas de auditoría pueden mejorar su competitividad, es desplegar y utilizar sus recursos primarios (auditores) de manera más efectiva. Dentro de este contexto, la optimización del proceso de auditoría en campo juega un papel importante en la planificación anual de recursos humanos de la firma, siendo para ello esencial el uso de modelos de selección de rutas de auditoría.

El problema tras la tarea de programación del personal de una firma de auditoría incluye varios objetivos que son típicos de los problemas de asignación de

empleados. Las firmas de auditoría se enfrentan al complejo trabajo de agendar auditores para la ejecución de procesos de auditoría, con diferente duración, en múltiples localidades, y con diferentes ventanas de tiempo. La programación se vuelve más compleja a medida que la empresa necesita considerar problemas de la vida real para determinar un horario óptimo o eficiente (Dodin & Eliman, 1997). Específicamente, se necesita desarrollar rutas que, minimicen el tiempo de viaje sin valor agregado que, equilibren la carga de trabajo entre los auditores y que, consideren su horario laboral (Begur et al., 1997).

Para el desarrollo de estas rutas, en el presente estudio, se propone el uso del modelo del Problema del Vendedor Viajero, TSP de sus siglas en inglés (*Traveling Salesman Problem*). El problema del vendedor viajero es un método matemático bien conocido en la optimización de rutas. Su objetivo es obtener la mejor ruta entre un conjunto determinado de clientes y depósitos, para minimizar la distancia y el costo del viaje (Navab-Kashani et al., 2015).

De acuerdo con lo anterior, el objetivo de este estudio es desarrollar un modelo, basado en el problema del vendedor viajero, que permita optimizar la elección de las rutas de auditorías para uno de los clientes de una empresa de certificación en Ecuador.

1.1 Descripción del problema

La empresa sujeto del presente estudio ofrece dentro de su cartera de servicios, con la ejecución de auditorías de inocuidad alimentaria a franquicias de restaurantes. Cuenta con una plantilla de cuatro auditores a tiempo completo en la ciudad de Guayaquil, donde está su sede, y tres auditores contratados por la prestación de servicios profesionales en las ciudades de Quito y Guayaquil, con quienes atiende todos los servicios nacionales de su cartera, incluyendo las auditorías a cinco franquicias multinacionales con aproximadamente 220 locales en territorio ecuatoriano.

La empresa en cuestión debe realizar, anualmente, auditorías a los diferentes locales de cada franquicia. Estas auditorías se realizan durante todo el año, debiendo

ser necesario disponer de los auditores para actividades adicionales como: entrenamientos, calibraciones, y otros servicios prestados por la compañía.

La compañía auditora atiende franquicias, con locales ubicados en diferentes ciudades y provincias en el Ecuador, con una duración de auditoría que varía entre dos y cuatro horas dependiendo del tamaño del local, y del tipo de protocolo a auditar. Los auditores usan el servicio de taxis, para sus movimientos locales dentro de cualquier ciudad donde se cuente con restaurantes; y para auditorías fuera de las ciudades de Quito y Guayaquil, se planifican viajes semanales que consideran el uso en transporte público para la movilidad entre ciudades. Cabe indicar que la jornada laboral legal en Ecuador es de lunes a viernes con 8 horas laborables (40 semanales), fuera de los horarios de alimentación.

Actualmente, se planifican las auditorías de manera semanal, y en mayor parte la selección de los locales a visitar por jornada depende del criterio del auditor. Sin embargo, para la planificación de las rutas de las auditorías se juzgó, que el empleo de reglas de sentido común y planificación aleatoria no permite el uso eficiente de los recursos, por lo que la aplicación de metodologías de optimización resulta de gran expectativa, para la compañía.

Se observa en la Tabla 1.1. un ejemplo de jornada de auditoría de una semana de duración, planificada por uno de los auditores de la compañía, siendo evidente que existe tiempo no utilizado que puede ser aprovechado.

Luego de conversaciones con los responsables de la planificación de actividades en la empresa sujeto de estudio, se propuso como parte del presente trabajo, la optimización de las rutas de auditorías, para los locales fuera de las ciudades de Quito y Guayaquil, de una de las cinco franquicias de alimentación a quienes se les entrega este servicio. Para efectos de esta tesis, se denominará a la franquicia como *Fast Food Inc.*, y su selección corresponde a que representa aproximadamente el 65% de locales de alimentación auditados por la empresa (144), generando una proporción significativa de su facturación total. Los locales parte de este estudio, son 48, ubicados en 27 ciudades, de 17 provincias en Ecuador, según se puede apreciar en la Tabla 1.2.

Tabla 1.1 Ejemplo de una jornada semanal de auditorías (Elaboración propia)

Día	Ciudad	Local	Duración (min)	Distancia (min)
1	Quito	FF-80-Quito	120	212
2	Riobamba	FF-133-Riobamba	120	20
2	Riobamba	FF-38-Riobamba	180	87
3	Baños de Agua Santa	FF-103-Baños	180	61
3	Ambato	FF-24-Ambato	180	20
4	Ambato	FF-77-Ambato	120	20
5	Ambato	FF-67-Ambato	120	55
5	Latacunga	FF-20-Latacunga	120	121
Total (h)				28.9

Tabla 1.2 Distribución de los locales en estudio (Elaboración propia)

País	Provincia	Ciudad	Locales
Ecuador	Azuay	Cuenca	5
Ecuador	Carchi	Tulcán	1
Ecuador	Chimborazo	Riobamba	2
Ecuador	Cotopaxi	Latacunga	1
Ecuador	El Oro	Machala	3
Ecuador	Esmeraldas	Esmeraldas	1
Ecuador	Esmeraldas	Atacames	1
Ecuador	Guayas	Milagro	1
Ecuador	Guayas	Daule	1
Ecuador	Guayas	Villamil Playas	1
Ecuador	Imbabura	Ibarra	3
Ecuador	Loja	Loja	2
Ecuador	Los Ríos	Quevedo	2
Ecuador	Manabí	Vinces	1
Ecuador	Manabí	Ventanas	1
Ecuador	Manabí	Babahoyo	2
Ecuador	Manabí	B. Caráquez	1
Ecuador	Manabí	Manta	4

País	Provincia	Ciudad	Locales
Ecuador	Manabí	Jipijapa	1
Ecuador	Manabí	Portoviejo	2
Ecuador	Pichincha	Cayambe	1
Ecuador	Santa Elena	Salinas	1
Ecuador	Sto. Domingo	Sto. Domingo	4
Ecuador	Tungurahua	Ambato	3
Ecuador	Tungurahua	Baños	1
Ecuador	Orellana	El Coca	1
Ecuador	Sucumbíos	Lago Agrio	1

El tiempo de auditoría para los locales seleccionados, se estableció con la guía de los auditores que los han visitado previamente en la prestación de este servicio, como se muestra en la lista de locales incluida en el Anexo 1 de este documento.

Considerando las restricciones de: jornada laboral de los auditores, tiempo de movilización entre locales y cumpliendo con las restricciones de duración mínima de auditoría que imponen los dueños de la franquicia, la pregunta es: ¿Qué configuración de las rutas por seguir, garantiza que las auditorías del grupo de locales seleccionados de la franquicia *Fast Food Inc.* se cumplan en el menor tiempo, y en cumplimiento del conjunto de requerimientos?

1.2 Requerimientos

Los requerimientos relacionados a la actividad de auditoría planteada son:

- Todos los locales deben ser visitados una vez durante el trimestre, y se debe cumplir el tiempo mínimo de permanencia del auditor en el local para la ejecución de la actividad de auditoría.
- Las rutas podrán tener cualquier duración deseada para garantizar el mínimo tiempo de traslado, siempre que esto no interfiera con el resto de los requerimientos.

- Los auditores deben realizar jornadas compatibles con la duración de la semana laboral legal establecida en Ecuador, esta es de 8 horas diarias y 40 horas semanales.
- Los desplazamientos hacia los locales se realizan desde una ciudad de origen, que puede ser Quito o Guayaquil, a la que se debe retornar una vez culminada la jornada de auditorías.
- Los desplazamientos dentro de una misma ciudad se consideran como fijos en todos los casos, con un valor de 20 minutos.

1.3 Hipótesis de Investigación

“Es posible optimizar la planificación de un conjunto de auditorías en una empresa de certificaciones, haciendo uso de una variante del problema del vendedor viajero (TSP)”.

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar un modelo de optimización basado en el problema del vendedor viajero, que permita la elección de rutas para las auditorías a los locales de la franquicia *Fast Food Inc.* ubicados fuera de las ciudades de Quito y Guayaquil en Ecuador, usando un software de optimización como herramienta de resolución.

1.4.2 Objetivos específicos

- Revisar la literatura actualizada en torno al modelo TSP y otros problemas de ruteo.
- Determinar la variante del modelo TSP, que defina de mejor manera el problema de asignación de auditorías para la empresa de certificaciones en Ecuador sujeto del estudio.
- Desarrollar un modelo para la optimización del problema de asignación de auditorías planteado, y ejecutarlo en un software de optimización.
- Discutir sobre la efectividad del modelo en la resolución del problema.

1.5 Plan de acción

Dada la presumible condición de NP-difícil del problema planteado, y con el objetivo de minimizar la carga computacional de búsqueda de rutas entre los 48 locales de la franquicia *Fast Food Inc.* seleccionados, se realizará la clusterización de locales en dos subconjuntos: Circuito Norte, y Circuito Sur del país, donde los locales a su vez estarán agrupados por ciudad. Se considera que auditores con sede en la ciudad de Guayaquil atienden el Circuito Sur, y que auditores con sede en Quito atienden el Circuito Norte. La asignación de los locales que pertenecerán a cada circuito dependerá de la distancia entre de la ciudad en que se encuentran, y las ciudades de Guayaquil o Quito. Se puede apreciar en la **Tabla 1.3.** la distribución de locales en el Circuito Norte, que abarca principalmente provincias de la región sierra ecuatoriana, pero también de las regiones costa y oriente; y en la **Tabla 1.4.** se aprecia la distribución de locales en el Circuito Sur, que abarca la mayor parte de las provincias de la costa ecuatoriana, y dos provincias de la región sierra. Se puede observar adicionalmente, la representación geográfica de ambos circuitos en la **Figura 1.2.**, donde se muestra además las ciudades de origen de cada uno de los recorridos.

Una vez realizada la división de los locales, se identificará y desarrollará el modelo matemático de optimización, aplicable al problema planteado y sus restricciones. El mismo que se utilizará paralelamente en los circuitos Norte y Sur, con la data de cada circuito. El modelo considerará un auditor por ciudad de origen, en búsqueda de una ruta óptima, de uno o varios ciclos, para visitar en el menor tiempo posible a los locales franquiciados correspondientes a cada circuito; y respetando las condiciones mencionadas en los requerimientos. Se espera obtener como resultado, una ruta de varios ciclos que recorra todos los locales del circuito en el menor tiempo posible. Se entiende por ciclo a una ruta con una duración máxima de 40 horas, incluyendo tiempo de auditoría y tiempo de movilización, que recorre un subconjunto de los locales de circuito, partiendo de una de las ciudades base, y retornando a la misma.

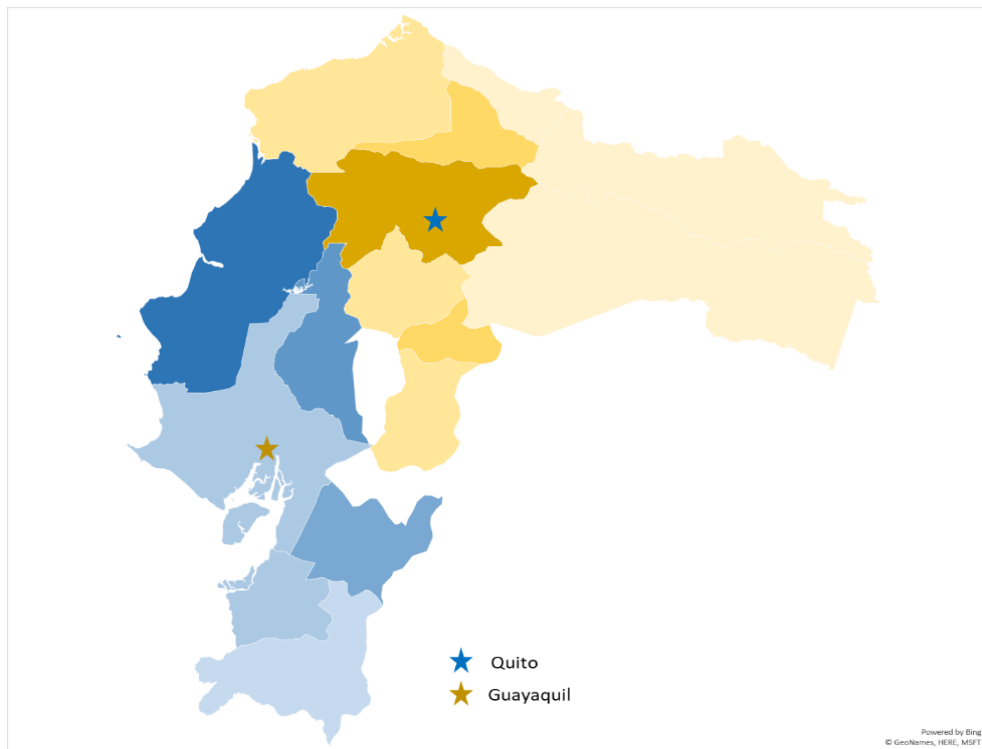


Figura 1.1 Representación de los Circuitos Norte y Sur (Elaboración propia)

Tabla 1.3 Distribución de los locales del Circuito Norte (Elaboración propia)

País	Provincia	Ciudad	Locales
Ecuador	Carchi	Tulcán	1
Ecuador	Chimborazo	Riobamba	2
Ecuador	Cotopaxi	Latacunga	1
Ecuador	Esmeraldas	Esmeraldas	1
Ecuador	Esmeraldas	Atacames	1
Ecuador	Imbabura	Ibarra	3
Ecuador	Pichincha	Cayambe	1
Ecuador	Sto. Domingo	Sto. Domingo	4
Ecuador	Tungurahua	Ambato	3
Ecuador	Tungurahua	Baños	1
Ecuador	Orellana	El Coca	1
Ecuador	Sucumbíos	Lago Agrio	1

Tabla 1.4 Distribución de los locales del Circuito Sur

País	Provincia	Ciudad	Locales
Ecuador	Azuay	Cuenca	5
Ecuador	El Oro	Machala	3
Ecuador	Guayas	Milagro	1
Ecuador	Guayas	Daule	1
Ecuador	Guayas	Villamil Playas	1
Ecuador	Loja	Loja	2
Ecuador	Los Ríos	Quevedo	2
Ecuador	Manabí	Vinces	1
Ecuador	Manabí	Ventanas	1
Ecuador	Manabí	Babahoyo	2
Ecuador	Manabí	B. Caráquez	1
Ecuador	Manabí	Manta	4
Ecuador	Manabí	Jipijapa	1
Ecuador	Manabí	Portoviejo	2
Ecuador	Santa Elena	Salinas	1

CAPÍTULO 2

Marco teórico

El problema del vendedor viajero es uno de los problemas más estudiados en la Investigación de Operaciones. Se centra en estudiar un problema de la siguiente clase: un vendedor debe visitar varios clientes y desea conocer cuál es el camino de mínima distancia que, partiendo de un origen, recorra a todas las ciudades y regrese al mismo. Este problema de apariencia sencilla es famoso por su gran complejidad computacional y pertenece a la categoría NP-difícil. A la actualidad, no se ha encontrado ningún algoritmo que logre resolverlo en un tiempo polinomial en instancias complejas o grandes. No obstante, su importancia no se debe únicamente a la complejidad de su resolución sino, a la gran variedad de situaciones prácticas, en las que pueden ser aplicados (Calviño, 2011).

2.1 Historia del TSP

La historia del TSP es larga y presentada por (Klansek, 2011), que señala: “Uno de los primeros problemas similares a TSP fue estudiado por Leonhard Euler en 1759, cuyo interés estaba en resolver el problema del recorrido del caballo de ajedrez”. Una solución precisa del problema del recorrido del caballo tendría a un caballo visitar cada uno de los 64 cuadrados de un tablero de ajedrez exactamente una vez en su recorrido y sin repetir. La primera formulación descriptiva del TSP, se puede encontrar en un manual alemán para vendedor viajero de Voigt en 1832. El manual mencionado considera la ruta más corta a lo largo de 45 ciudades alemanas. Teniendo en cuenta las condiciones de viaje de esa época, la propuesta de la ruta alemana podría incluso ser óptima. El matemático austríaco Karl Menger parece ser el primer investigador en escribir la formulación matemática del TSP. Los resultados de su trabajo, de investigación pionero en el TSP se publicaron en 1932.

En 1937, Merrill Flood de la Universidad de Columbia presentó, la programación óptima de rutas para autobuses escolares resolviendo el TSP. Una contribución significativa fue de George Dantzig, Delbert Fulkerson y Selmer Johnson en 1954, quienes expresaron el TSP como una programación lineal de enteros mixtos

para problemas de optimización (MILP) y desarrollaron, el método del plano de corte. Mostraron la efectividad de su método de optimización, resolviendo el TSP de 49 ciudades. El trabajo de investigación presentado, por Dantzig en 1954, representa un hito importante en el histórico desarrollo, de los códigos de computadora para resolver el TSP.

2.2 Modelado y formulación del TSP

El problema del agente viajero, aplicado al caso presentado de auditorías a locales de la franquicia *Fast Food Inc.*, tiene un conjunto de n locales que denominamos nodos y un conjunto de rutas entre locales denominados arcos. Sea $G = (V, A)$ un grafo completo, donde $V = (1 \dots n)$ es el conjunto de nodos, y A el conjunto de arcos que conectan esos nodos, a los que se asocia un valor no negativo C_{ij} , que es la distancia recorrida por el auditor para trasladarse del local i al local j . Si la distancia C_{ij} es igual a la distancia C_{ji} , se dice que estamos tratando con un problema simétrico, o STSP, caso contrario, el problema es considerado como asimétrico o ATSP.

El objetivo principal del problema es encontrar la ruta más corta, posible que visite exactamente una vez, cada local y luego retorne a su origen (Hincapié et al., 2004). Si definimos las variables dicotómicas de decisión x_{ij} para todo $(i, j) \in A$, de forma que tomen el valor 1 si el arco (i, j) forma parte de la solución y cero en el caso opuesto.

El TSP puede ser formulado como un problema de la programación lineal en enteros, donde la función objetivo es minimizar la sumatoria de las distancias recorridas. De tal forma el modelo matemático se puede representar de la siguiente forma:

$$\text{Min} \sum_i^n \sum_j^n C_{ij} X_{ij} \quad (1)$$

Donde,

$$X_{ij} = \begin{cases} 0, & \text{si se recorre el arco de } i \text{ a } j \\ 1, & \text{de lo contrario} \end{cases} \quad (2)$$

Sujeto a:

$$\sum_{i=0, i \neq j}^n X_{ij} \quad j = 0, \dots, n \quad (3)$$

$$\sum_{j=0, j \neq i}^n X_{ij} \quad i = 0, \dots, n \quad (4)$$

+ Restricciones de eliminación de subtours

La ecuación (2) indica la condición de binaria de la variable X, que obliga a tomar la decisión de elegir o no recorrer el arco entre un nodo y otro. Las ecuaciones (3) y (4) aseguran que el auditor sale únicamente de un nodo, y que un nodo recibe únicamente a un auditor. Ambas restricciones también implican que exista exactamente una salida desde el local de origen, y obligan a que un solo camino cubra todos los locales y no dos o más caminos.

Además de las restricciones ya mencionadas, es necesaria la implementación de restricciones para la eliminación de subtours, las mismas que aportan a la complejidad computacional del problema, al incrementar exponencialmente el número de restricciones en función del número de nodos. Entre las principalmente abordadas en la literatura para la eliminación de subtours son: La formulación Danzig-Fulkerson-Johnson (DFJ)(5), la formulación Miller -Tucker-Zemlin (MTZ)(6), y la formulación Desrochers-Laporte (DL)(7), que es una variante más eficiente de la formulación MTZ (Öncan et al., 2009).

$$\sum_{i,j \in S} X_{ij} \leq |S| - 1 \quad (S \subset V, \quad 2 \leq |S| \leq n - 2) \quad (5)$$

$$\begin{aligned} U_i - U_j + nX_{ij} &\leq n - 1 & i, j &= 2..n \\ 1 \leq U_i &\leq n - 1 & i &= 2..n \end{aligned} \quad (6)$$

$$\begin{aligned} U_i - U_j + (n - 1)X_{ij} + (n - 3)X_{ji} &\leq n - 2 & i, j &= 2..n \\ 1 \leq U_i &\leq n - 1 & i &= 2..n \end{aligned} \quad (7)$$

De las anteriores, la formulación DFJ presenta la mayor complejidad computacional en problemas grandes, al incrementar el número de restricciones del problema en $2^{n-1} + n - 2$, comparado con las $n^2 - n + 2$ restricciones generadas por las formulaciones de MTZ y DL. Sin embargo, la técnica de generación de restricciones, basada en la formulación DFJ resulta más atractivo al momento de tratar problemas de mayores dimensiones (Behhida & Mir, 2018). Esta técnica consiste en resolver el modelo TSP sin restricciones de eliminación de subtours, y de generarse estos en la respuesta, incluir en la formulación las restricciones correspondientes a los subtours identificados, repitiendo esto hasta generar la respuesta óptima factible.

2.3 Variantes del TSP

Existen muchas variantes al problema de vendedor viajero, descrito anteriormente. Seguidamente se enumeran algunas de ellas:

- MAX-TSP.- Consiste en encontrar un circuito hamiltoniano de costo máximo.
- TSP con cuello de botella.- Consiste en encontrar un circuito hamiltoniano tal que minimice el mayor costo de entre todas las aristas de este, en vez de minimizar el costo total.
- TSP gráfico.- Consiste en encontrar un circuito de costo mínimo tal que se visiten los locales al menos una vez.
- TSP agrupado.- Los nodos o locales están divididos en "clústeres" o grupos, de manera que lo que se busca es un circuito hamiltoniano de costo mínimo en el que se visiten los nodos de cada grupo de manera consecutiva.
- TSP generalizado.- Los nodos o locales también están divididos en grupos, pero lo que se busca es un circuito de costo mínimo que visite exactamente un nodo de cada grupo.
- TSP con múltiples viajantes. - Existen un número m de viajantes, cada uno de los cuales debe visitar algunos de los locales. El problema se transforma, por tanto, en la búsqueda de una partición de los nodos a visitar X_1, \dots, X_m y de m ciclos, uno para cada X_i , de manera que la suma de las distancias recorridas por los m auditores sea mínima. Esta variante puede ser vista también como una simplificación de los problemas de rutas de vehículos.

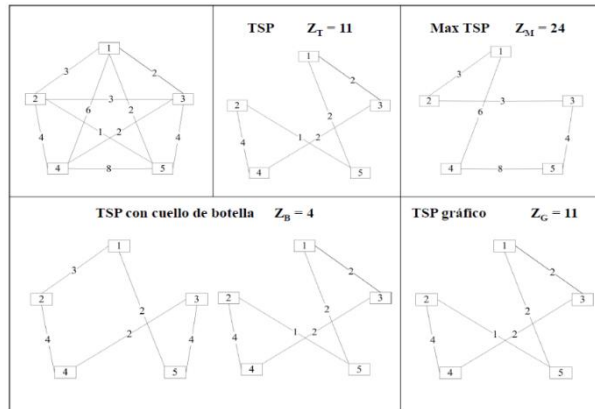


Figura 2.1 Variantes del TSP: Máximo, cuello de botella y gráfico. (Calviño, 2011).

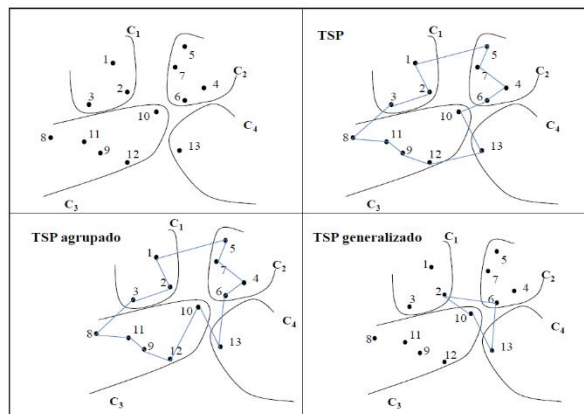


Figura 2.2 Variantes del TSP: Agrupado y Generalizado. (Calviño, 2011).

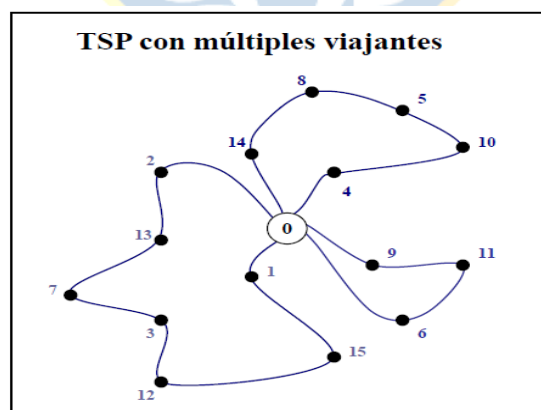


Figura 2.3 Variantes del TSP: Múltiples Viajantes. (Calviño, 2011).

CAPÍTULO 3:

Estado del arte

El problema del vendedor viajero (TSP) es uno de los problemas de optimización combinatorial NP-Duro más citados porque es muy fácil de entender, pero difícil de resolver. Es un problema desafiante de valor académico significativo ya que a menudo se usa como un problema de referencia cuando se desarrollan nuevos enfoques de solución (Liu, 2014).

El objetivo es encontrar un recorrido que minimice el costo (generalmente la distancia total), entre todos los recorridos que satisfacen este criterio. El problema se puede visualizar en un gráfico. Cada ciudad se convierte en un nodo. Las longitudes de los bordes corresponden a la distancia entre las ciudades adjuntas. Entonces, TSP puede formularse como encontrar un ciclo hamiltoniano con la longitud mínima (Cicková et al., 2008).

Como un problema NP-difícil, TSP siempre ha sido valorado por los investigadores. En la investigación sobre TSP, la mayoría de los problemas se tratan como un problema de un solo objetivo, o se consideran desde dos perspectivas separadas. Para este último, generalmente se utilizan dos funciones objetivas: minimizar la distancia total recorrida y minimizar la distancia de viaje del viajero más largo. El segundo objetivo suele ser una condición para equilibrar los subtours, lo que implica algunos problemas, como la carga de trabajo entre los vendedores y el tiempo de servicio de cada cliente en una situación práctica. Por lo tanto, optimizar la distancia total y el equilibrio entre subtours son dos objetivos en conflicto que determinan que no pueda ser considerado por separado (Shuai et al., 2019).

3.1 Solucionadores no exactos

Estos solucionadores ofrecen soluciones potencialmente no óptimas, pero típicamente más rápidas. En cierto modo, la compensación opuesta de los solucionadores exactos. Los solucionadores no exactos se pueden subdividir en (Kotthoff et al., 2015) (Chauhan et al., 2012).

- Algoritmos de aproximación: estos algoritmos vienen con el peor factor de aproximación para la solución encontrada. Los dos métodos tradicionales para resolver el TSP son un algoritmo basado en MST puro, que logra una aproximación del factor 2 y un algoritmo basado en el MST y el Problema Mínimo de Coincidencia (MMP) debido a Christofides, que logra una aproximación del factor $3/2$. Ambos métodos están restringidos al MTSP ya que dependen de la desigualdad del triángulo. El PTAS para Euclidean TSP es principalmente un resultado teórico debido a su tiempo de ejecución prohibitivo.
- Algoritmos heurísticos: estos algoritmos solo prometen una solución factible. Van desde métodos simples de construcción de recorridos como *Nearest Neighbor*, *Clarke-Wright* y *Multiple Fragment* hasta algoritmos de mejora de recorridos más complicados como *Tabu Search* y *Lin-Kernighan*. Finalmente, hay un grupo de algoritmos fascinantes que desafortunadamente tienden a combinar soluciones aproximadas y grandes tiempos de ejecución. Aquí encontramos métodos como *recocido simulado*, algoritmos genéticos, algoritmos de colonias de hormigas y algoritmos de aprendizaje automático como redes neuronales (Liu, 2014) (Cicková et al., 2008) (Kamarudin et al., 2016).

Una forma de clasificar algoritmos es por principio de implementación (Zelinka, 2002):

- Enumeración explícita: Conduce al reconocimiento de todas las posibles soluciones de problemas, por lo tanto, es aplicable solo para problemas pequeños.
- Métodos deterministas: Estos algoritmos se basan solo en métodos rigurosos de matemática "clásica". Alguna información adicional, como gradiente, convexidad, etc., generalmente se necesita (Algoritmo de ramificación y límite, Método de plano de corte, Programación dinámica, etc.).
- Métodos estocásticos: Esos algoritmos funcionan en métodos probabilísticos para resolver problemas. Los algoritmos estocásticos

funcionan lenta y generalmente son aplicables solo para "adivinar" (Monte Carlo, caminata de búsqueda aleatoria, cálculo evolutivo, etc.).

- Métodos combinados: Los métodos combinados están compuestos por composición estocástica y determinista. Se han ideado varios algoritmos de metaheurística (optimización de colonias de hormigas, algoritmos miméticos, algoritmos genéticos, etc.).

3.2 Solucionadores exactos

Hay dos grupos de solucionadores exactos. Una de ellas es resolver las relajaciones de la formulación de la Programación lineal de TSP y utiliza métodos como Plano de corte, Punto interior, Ramificación y unión y Ramificación y corte. Otro grupo más pequeño está utilizando la programación dinámica. Para ambos grupos, la característica principal es la garantía de encontrar soluciones óptimas a expensas del tiempo de ejecución y los requisitos de espacio (Chauhan et al., 2012).

- Ramificación y Unión: enumera implícitamente todas las soluciones factibles, usando cálculos donde las restricciones de los problemas se relajan. En otras palabras, la estrategia de ramificación y de división divide un problema a resolver en varios subproblemas. Es un sistema para resolver una secuencia de subproblemas, cada uno de los cuales puede tener múltiples soluciones posibles y donde la solución elegida para un subproblema puede afectar las posibles soluciones de subproblemas posteriores.
- El Plano de Corte: es un método que refina iterativamente un conjunto factible o una función objetivo mediante desigualdades lineales, denominadas cortes. Tales procedimientos se usan comúnmente para encontrar soluciones enteras a problemas de programación lineal de enteros mixtos (MILP), así como para resolver problemas generales de optimización convexa no necesariamente diferenciables.
- Ramificación y Corte: Este método resuelve el programa lineal sin la restricción de entero utilizando el algoritmo simplex regular. Cuando se obtiene una solución óptima, y esta solución tiene un valor no entero para una variable que se supone que es entera, se utiliza un algoritmo de plano de corte

para encontrar restricciones lineales adicionales que se satisfacen con todos los puntos enteros factibles pero violados por la corriente solución fraccionada. Si se encuentra tal desigualdad, se agrega a la formulación, de modo que resolverla producirá una solución diferente que es de esperar "menos fraccional". Este proceso se repite hasta que se encuentre una solución entera (que luego se sabe que es óptima) o hasta que no se encuentren más planos de corte.

- Programación Dinámica: es una técnica para calcular eficientemente las recurrencias almacenando resultados parciales y reutilizándolos cuando sea necesario. Es bien sabido que las recurrencias de programación dinámica pueden expresarse como problemas de ruta más corta en una red en capas.
- Método de Fuerza Bruta: cuando uno piensa en resolver TSP, el primer método que puede venir a la mente es un método de fuerza bruta. El método de fuerza bruta es simplemente generar todos los recorridos posibles y calcular sus distancias. El recorrido más corto es, por lo tanto, el recorrido óptimo.

En la Tabla 3.1 se puede observar algunos de los principales hitos en el desarrollo de soluciones al TSP.

Tabla 3.1 Hitos en el desarrollo del TSP (Chauhan et al., 2012)

Año	Hito	Autor(es)
1954	49-point instance solved by LP and by adding cutting planes manually.	Dantzig, Fulkerson and Johnson
1970	Lagrangian relaxation. Error about 1%.	Held and Karp
1973	k-Opt heuristic. 1% to 2% above optimal.	Lin and Kernighan
1976	1:5-approximation.	Christodes
1983	Simulated annealing-based heuristic.	Kirkpatrick, Gelatt and Vecchi
1985	Recurrent neural network-based heuristic.	Hopfield and Tank
1992	TSP heuristics by using k-d trees.	Bentley
1995	7,392-point instance solved by LP and cutting planes generation (Concorde).	Applegate, Bixby, Chvatal and Cook
1996	PTAS for the Euclidean TSP. $nO(1/\epsilon)$ time.	Arora
1998	Improved k-opt heuristic (LKH). Within 1% above optimal.	Helsgaun
2004	24,978-point instance solved by LKH and proved by Concorde.	Applegate, Bixby, Chvatal, Cook and Helsgaun
2006	85,900-point instance solved by Concorde.	Applegate, Bixby, Chvatal, Cook, Espinoza, Goycoolea and Helsgaun

CAPÍTULO 4:

Metodología

4.1 Definición de los circuitos

Como se mencionó en el Plan de Acción del Capítulo 1, con el objetivo de facilitar la resolución del problema, los locales a visitar se dividieron en dos conjuntos: Circuito Norte, y Circuito Sur. Dentro de estos conjuntos, a su vez, se agruparon los locales por ciudad, adoptando un enfoque de TSP Generalizado.

Para el Circuito Norte, la agrupación de sus 20 se realizó en función de las 12 ciudades en que se encuentran, mientras que los 28 locales correspondientes al Circuito Sur se agruparon en las 15 ciudades donde se encuentran ubicados (ver Tabla 4.1).

Las matrices de distancia para cada circuito se desarrollaron con la ayuda del software de geolocalización *Google Maps*, utilizando como valor el tiempo estimado por el software para la movilización entre ciudades los pares de ciudades, sin considerar aspectos como la variabilidad por horario o día de la semana.

El código de local se ha establecido en función de su orden de su apertura. Dado que para este trabajo no se han considerado los locales ubicados en las ciudades de Quito y Guayaquil, se observa que la numeración de los locales seleccionados no es consecutiva.

Para ciudades que cuentan con varios locales, el tiempo de auditoría en el nodo se estableció como la suma de los tiempos de auditoría de cada local, más un tiempo adicional de 20 minutos por cada local después del primero, considerado como el tiempo promedio de movilización entre locales fuera de las ciudades de Quito y Guayaquil.

Los Anexos 2 y 3 muestran las matrices de distancias y tiempos de auditoría utilizadas para la optimización de cada uno de circuito los circuitos obtenidos por la clusterización de locales.

Tabla 4.1 Distribución de los locales del Circuito Norte (Elaboración propia)

Circuito	Ciudad	Local(es)
Norte	Tulcán	FF-107
	Riobamba	FF-38, FF-133
	Latacunga	FF-20
	Esmeraldas	FF-57
	Atacames	FF-71
	Ibarra	FF-62, FF-69, FF-113
	El Coca	FF-106
	Cayambe	FF-68
	Santo Domingo	FF-39, FF-83, FF-121, FF-123
	Lago Agrio	FF-74
	Ambato	FF-24, FF-67, FF-77
	Baños	FF-103
	Sur	
Cuenca		134
Machala		FF-49, FF-50, FF-104
Daule		FF-36
Milagro		FF-23
Villamil		FF-40
Loja		FF-27, FF-79
Babahoyo		FF-11, FF-37
Quevedo		FF-52, FF-61
Ventanas		FF-105
Vinces		FF-98
B. Caráquez		FF-73
Jipijapa		FF-100
Manta		FF-90, FF-95, FF-126
Portoviejo		FF-129, FF-135
Salinas		FF-141

4.2 Modelo matemático

Dada la configuración del problema, se busca que uno o varios auditores recorran un conjunto de ciudades respetando las características del servicio explicadas en el Capítulo 1, siendo las principales:

- Respetar la jornada laboral de 40 horas semanales de los auditores,
- Cumplir con los tiempos mínimos de auditoría en cada local

Para esto, la variante del TSP seleccionada para la modelación matemática del presente problema, es SD-MTSP (*Single Depot Multiple Traveling Salesman Problem*), que puede considerarse como una relajación del Problema de Ruteo de Vehículos VRP, donde no se incluyen restricciones de capacidad (Bektas, 2006). En este se busca minimizar tanto el recorrido total del conjunto de vendedores para visitar todos los nodos, como la diferencia entre los recorridos de los vendedores (buscando una distribución equitativa de la carga de viaje) (Shuai et al., 2019).

En el caso particular del problema planteado, se considera que las múltiples rutas que debe arrojar el modelo en su solución corresponden al número de ciclos que cumplen con la restricción de la duración máxima de la jornada laboral de los auditores. De esta manera, dado un conjunto de n número de nodos, expresados como $L = \{i\}, i = 1, 2, \dots, n$; y un conjunto de m elementos que representa el número de ciclos, representados como $C = \{k\}, k = 1, 2, \dots, m$, y que puede ser una constante, o una variable entera acotada (Gavish & Srikanth, 1986); se considera que la distancia entre el local i y el local j se representa como d_{ij} , y el tiempo de auditoría es de cada local como T_i . La variable X_{ijk} toma el valor de 1 si el auditor toma la ruta del local i al local j durante el ciclo k , caso contrario toma el valor de cero.

Entonces, cada nodo será visitado exactamente una vez por el auditor, excepto por el punto de partida, al que se regresará un máximo de m veces. Para la formulación del problema SD-MTSP que resuelva el problema presentado, se usan los siguientes parámetros y variables:

Variables:

X_{ijk} : Variable binaria que indica la selección de una ruta durante un ciclo

U_i : Número de nodos visitados desde el origen al nodo i . (parte de las restricciones de eliminación de subtours).

Parámetros:

C_{ij} : Distancia expresada en minutos entre el nodo i y j .

T_i : Tiempo de auditoría en el nodo i , correspondiente a la suma de los tiempos de auditoría en todos los locales del nodo (ciudad), más los tiempos de traslados entre los locales.

n : Número de nodos del circuito.

m : Número de ciclos del circuito, obtenidos de una solución inicial del problema, mediante una heurística Greedy.

El modelo matemático se detalla como:

$$\text{Min} = \sum_{k=1}^m \sum_{i=0}^n \sum_{j=0}^n C_{ij} * X_{ijk} + \sum_{i=0}^n T_i * X_{ijk} \quad (8)$$

Donde,

$$X_{ijk} = \begin{cases} 1, & \text{si e auditor pasa del local } i \text{ al } j \text{ en el ciclo } k \\ 0, & \text{de otro modo} \end{cases} \quad (9)$$

Sujeto a:

$$\sum_{k=1}^m \sum_{i=1}^n X_{ijk} = 1; \quad \forall k = 1, \dots, m, \quad \forall j = 2, \dots, n \quad (10)$$

$$\sum_{k=1}^m \sum_{j=1}^n X_{ijk} = 1; \quad \forall k = 1, \dots, m, \quad \forall i = 2, \dots, n \quad (11)$$

$$\sum_{i=1}^n X_{i1k} \leq 1; \quad \forall k = 1, \dots, m, \quad (12)$$

$$\sum_{j=1}^n X_{1jk} \leq 1; \quad \forall k = 1, \dots, m, \quad (13)$$

$$\sum_{k=1}^m \sum_{i=1, i \neq j}^n X_{i1k} \leq m; \quad (14)$$

$$\sum_{k=1}^m \sum_{j=1, j \neq i}^n X_{1jk} \leq m; \quad (15)$$

$$\sum_{i=0}^n \sum_{j=0}^n C_{ij} * X_{ijk} + \sum_{i=0}^n T_i * X_{ijk} \leq 2400; \quad i \neq j, \forall k = 1, \dots, m \quad (16)$$

$$\sum_{i=1}^n X_{ijk} - \sum_{j=1}^n X_{jik} = 0 \quad i \neq j, \forall k = 1, \dots \quad (17)$$

+ Restricciones de eliminación de subtours

La ecuación (8) expresa la función objetivo del problema, que busca minimizar el tiempo de recorrido y auditoría en los n nodos parte de un circuito, y en m ciclos; arrojando, al incorporar la dimensión k , una respuesta de tamaño n por $(n*m)$, de mayor dimensión que la solución de un problema MTSP tradicional, de n por n . El objetivo es abordar independientemente cada ciclo para garantizar el cumplimiento de la característica de duración de jornadas del problema, expresada en la ecuación (16). En la ecuación (9) se establece la condición binaria de la variable X_{ijk} , que toma el valor de 1 si el auditor se moviliza del nodo i al nodo j durante el ciclo k .

Las ecuaciones (10) y (11) garantizan que todos los nodos, excepto el origen, son visitados exactamente una vez al cabo de los m ciclos. En (12) y (13) se asegura la eliminación de múltiples rutas dentro de un ciclo, asegurando que por cada ciclo, máximo un recorrido parte del origen, y retorna al mismo.

Las expresiones (14) y (15) son una adaptación de las restricciones típicas del problema MTSP, que buscan que un número determinado de viajantes partan del origen y regresen al mismo. Para el caso de este modelo, establece que el número de ciclos en la solución (que cumpla con todas las restricciones del problema) es menor

o igual a m . Este valor se estima utilizando como parte de la metodología, una solución heurística inicial para el problema, y puede ser reducido en busca de la solución óptima. Finalmente, la restricción (17) establece que debe existir continuidad en los flujos dentro de cada ciclo, es decir, cada nodo en el que exista una salida debe también recibir una llegada.

Como estrategia de eliminación de subtours, el modelo matemático considerará el uso paralelo de tres formulaciones ampliamente tratadas en la literatura: las formulaciones DL (18), MTZ (19), incluidas en la sección de modelado del TSP en el Capítulo 2, y la formulación de (Gavish, 1976) GA (20), derivada de la formulación de (Svestka & Huckfeldt, 1973), y ajustada específicamente al el problema MTSP, restando en la ecuación el número de ciclos (vendedores) en el problema, al asumir que el número máximo de nodos que puede visitar un viajante es de $n-m$. Estas formulaciones se detallan en las ecuaciones a continuación:

$$U_i - U_j + (n - 1) * X_{ijk} + (n - 3) * X_{jik} \leq n - 2; \quad i \neq j, \forall j \\ = 2, \dots n \quad \forall k = 1, \dots m \quad (18)$$

$$U_i - U_j + nX_{ijk} \leq n - 2; \quad i \neq j, \forall i, j = 2..n, \forall k \\ = 1, \dots m \quad (19)$$

$$U_i - U_j + (n - m)X_{ijk} \leq n - m - 1; \quad i \neq j, \forall i, j = 2..n, \forall k \\ = 1, \dots m \quad (20)$$

4.2.1 Acotamiento de ciclos

Con el objetivo de reducir tiempos computacionales del modelo, las ecuaciones (15) y (16), que establecen la exploración de entre 1 y m ciclos para la obtención de la solución, se modifican a las expresiones (21) y (22), donde reducen la exploración a un rango de 2 (entre $m-1$ y m ciclos). Esta acotación busca también restringir la exploración de soluciones potencialmente no factibles, y bajo el supuesto de que si el óptimo global de los circuitos, obtenido mediante métodos exactos, es menor al de la

solución inicial, podría requerirse hasta un ciclo menos para que el conjunto de auditorías sea sean completado en cumplimiento de las demás restricciones.

$$m - 1 \leq \sum_{k=1}^m \sum_{i=1, i \neq j}^n X_{i1k} \leq m; \quad (21)$$

$$m - 1 \leq \sum_{k=1}^m \sum_{j=1, j \neq i}^n X_{1jk} \leq m; \quad (22)$$

4.3 Solución inicial

Para definir el número de ciclos m , requerido para que la solución del problema cumpla con la restricción de duración máxima de la jornada de los auditores, se obtiene una solución inicial mediante una heurística de tipo *Nearest Neighbor* (NN), cuyo pseudocódigo se observa en la tabla 4.2. La elección de este tipo de heurística corresponde a que por sus características, imita la forma natural en que un auditor elegiría su ruta (Chauhan et al., 2012).

La solución inicial de este problema es un grafo dirigido que se obtiene partiendo del origen, y visitando el nodo más cercano mientras que $T + t_0$ sea menor o igual a J , es decir, que la suma del tiempo de movilización partiendo desde el origen, y recorriendo un conjunto de nodos, el tiempo de auditoría de estos nodos, y el tiempo de retorno desde el último nodo hasta el origen, no supere el tiempo máximo establecido para la jornada de auditoría. Si esto pasa, se vuelve a partir desde el origen, generando un nuevo ciclo de auditoría. Al final la solución arrojará una secuencia de recorrido, que abarque todos los locales, que cumpla con las restricciones planteadas, y el número m de ciclos requeridos para ello. Adicionalmente calculará el tiempo total de todos los ciclos.

Los parámetros considerados durante la solución heurística del problema son:

- n :** Número de nodos $\{C_0, C_1 \dots C_n\}$
- d :** Distancia de nodo i a nodo j , expresada en tiempo
- t :** Tiempo de auditoría por nodo

- T:** Suma de tiempos de jornada
J: Duración máxima de jornada
R: Conjunto de respuesta

Tabla 4.2 Algoritmo NN utilizado para la solución inicial (Elaboración propia)

ALGORITMO Solución inicial de rutas de auditoría;

ENTRADA: n, d, t, J, R, T ;
 $R = \{\emptyset\}$;
 $T = 0$;

INICIO

Hasta que $\{ C_0, C_1 \dots C_n \} \in R$, **Repetir:**

Incluir C_0 en R;
 Incluir en R al vecino v_i más cercano a v_0 ;
 $R = \{ C_0, C_i \}$;
 $T = d_{0i} + t_i$;

Si $T + t_0 \leq J$, **entonces:**
 Incluir en R al vecino C_j más cercano a C_i , tal que $C_j \notin R$;
 $T = T + d_{ij} + t_i$;

De otra forma:
 Parar;

Parar;

SALIDA: R, T

CAPÍTULO 5:

Resultados

5.1 Solución inicial

Como se detalló en el capítulo de metodología, el primer paso en la resolución del problema fue la generación de una solución, factible para todas las restricciones del problema, utilizando una heurística del tipo NN con los datos de cada uno de los circuitos ubicados en los Anexos; de manera en que se pueda obtener un dato estimado de la cantidad de ciclos requeridos para cumplir con las auditorías del circuito, tomando en consideración los tiempos máximos establecidos por jornada laboral semanal. Esta resolución se desarrolló de forma manual en una hoja de cálculo de Microsoft Excel. Los resultados obtenidos por cada uno de los circuitos, luego de la aplicación del método heurístico, se detallan en las secciones siguientes.

5.1.1 Resultados iniciales Circuito Norte

Este circuito comprende principalmente ciudades de región sierra ecuatoriana, junto a las ciudades de Esmeraldas y Atacames en la región costa; y, Lago Agrio y El Coca en la región oriental. El circuito cuenta con 20 locales, que luego de ser clusterizados en las ciudades a las que pertenecen, generaron 12 nodos, con entre uno y cuatro locales.

Todos los nodos del circuito se recorren tomando como punto de origen a la ciudad de Quito, desde donde parten los auditores. Luego de aplicado el método de solución inicial, se obtuvieron 3 ciclos de auditorías:

- **Ciclo #1:** Quito - Latacunga - Ambato - Baños - Riobamba - Quito.
- **Ciclo #2:** Quito - Cayambe - Ibarra - Tulcán - Lago Agrio - El Coca - Quito.
- **Ciclo #3:** Quito - Sto. Domingo - Esmeraldas - Atacames - Quito.

Tabla 5.1 Solución inicial Circuito Norte (Elaboración propia)

Ciclo	Ciudades visitadas	Locales visitados	Respuesta (min)	Tiempo (h)
1	4	7	1616	26.9
2	5	7	2178	36.3
3	3	6	1725	28.75
Total			5519	92.0

5.1.1 Resultados iniciales Circuito Sur

Este circuito comprende a la mayoría de las ciudades ubicadas región costa del Ecuador, junto a dos ciudades del sur de la región sierra (Cuenca y Loja); siendo visitadas desde la base en la ciudad de Guayaquil, donde se encuentra la oficina de la empresa auditora. El circuito cuenta con 28 locales, que luego de ser clusterizados en las ciudades a las que pertenecen, generaron 15 nodos, con entre uno y cinco locales.

Luego de aplicado el método de solución inicial, se obtuvieron 4 ciclos de auditorías:

- **Ciclo #1:** Guayaquil - Villamil - Salinas - Manta - B. Caráquez - Quevedo - Guayaquil.
- **Ciclo #2:** Guayaquil - Milagro - Babahoyo - Vinces - Daule - Jipijapa - Portoviejo - Guayaquil.
- **Ciclo #3:** Guayaquil - Ventanas - Machala - Cuenca - Guayaquil.
- **Ciclo #4:** Guayaquil - Loja - Guayaquil.

Tabla 5.2 Solución inicial Circuito Sur (Elaboración propia)

Ciclo	Ciudades visitadas	Locales visitados	Respuesta (min)	Tiempo (h)
1	5	9	2158	29.5
2	6	8	1767	36.0
3	3	9	2213	36.9
4	1	2	1116	18.6
Total			7254	120.9

Tabla 5.3 Matriz de solución inicial Circuito Norte (Elaboración propia)

	1	2	3	4	5	6	7	8	9	10	11	12	13
1.Quito	-	0	0	1	0	0	0	0	1	1	0	0	0
2.Tulcán	0	-	0	0	0	0	0	0	0	0	1	0	0
3.Riobamba	1	0	-	0	0	0	0	0	0	0	0	0	0
4.Latacunga	0	0	0	-	0	0	0	0	0	0	0	1	0
5.Esmeraldas	0	0	0	0	-	1	0	0	0	0	0	0	0
6.Atacames	1	0	0	0	0	-	0	0	0	0	0	0	0
7.Ibarra	0	1	0	0	0	0	-	0	0	0	0	0	0
8.El Coca	1	0	0	0	0	0	0	-	0	0	0	0	0
9.Cayambe	0	0	0	0	0	0	1	0	-	0	0	0	0
10.Sto. Domingo	0	0	0	0	1	0	0	0	0	-	0	0	0
11.Lago Agrio	0	0	0	0	0	0	0	1	0	0	-	0	0
12.Ambato	0	0	0	0	0	0	0	0	0	0	0	-	1
13.Baños	0	0	1	0	0	0	0	0	0	0	0	0	-

Tabla 5.4 Matriz de solución inicial Circuito Sur (Elaboración propia)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1.Guayaquil	-	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0
2.Cuenca	1	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.Machala	0	1	-	0	0	0	0	0	0	0	0	0	0	0	0	0
4.Daule	0	0	0	-	0	0	0	0	0	0	0	1	0	0	0	0
5.Milagro	0	0	0	0	-	0	0	1	0	0	0	0	0	0	0	0
6.Villamil	0	0	0	0	0	-	0	0	0	0	0	0	0	0	1	0
7.Loja	1	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0
8.Babahoyo	0	0	0	0	0	0	0	-	0	0	1	0	0	0	0	0
9.Quevedo	1	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0
10.Ventanas	0	0	1	0	0	0	0	0	0	-	0	0	0	0	0	0
11.Vinces	0	0	0	1	0	0	0	0	0	0	-	0	0	0	0	0
12.B. Caráquez	0	0	0	0	0	0	0	0	1	0	0	-	0	0	0	0
13.Jipijapa	0	0	0	0	0	0	0	0	0	0	0	0	-	0	1	0
Manta	0	0	0	0	0	0	0	0	0	0	0	1	0	-	0	0
Portoviejo	1	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0
Salinas	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-

5.2 Solución del modelo matemático

Para la prueba del modelo matemático desarrollado, se utilizó el programa de optimización LINGO 18.0, ejecutado en un CPU Intel® Core™ i5-8250 con 1.60-1.80 GHz, y 8 GB de memoria RAM, considerando como instancias al Circuito Norte con 13 nodos y el Circuito Sur con 16 Nodos, en ambos casos incluyendo el origen. En el Anexo 4, se observa el modelo matemático utilizado en Lingo.

5.2.1 Resultados circuito norte

Como resultado de la aplicación del modelo matemático desarrollado a la data del Circuito Norte, no se mejora significativamente el tiempo de auditoría obtenido durante la solución inicial (mejora de 0,07%). Las rutas obtenidas bajo el modelo tienen los mismos nodos que las obtenidas previamente, variando estas últimas únicamente en el orden en que se visitan los nodos:

Los ciclos obtenidos mediante el modelo son:

- **Ciclo #1:** Quito - Cayambe - Ibarra - Tulcán - Lago Agrio - El Coca - Quito.
- **Ciclo #2:** Quito - Esmeraldas - Atacames - Sto. Domingo - Quito.
- **Ciclo #3:** Quito - Latacunga - Ambato - Riobamba - Baños - Quito.

Tabla 5.5 Solución Circuito Norte (Elaboración propia)

Ciclo	Ciudades visitadas	Locales visitados	Duración de Ciclos (min)	Duración de Ciclos (h)
1	5	7	2178	26.9
2	3	6	1724	36.3
3	4	7	1613	28.75
F.O.			5515 min	91.9 h

La respuesta obtenida con el uso del modelo es el óptimo global, mejorando en una pequeña proporción a la solución heurística, y manteniendo la necesidad de 3 ciclos para cumplir con las auditorías a todos los locales de este circuito, con ciclos de en promedio $30,65 \pm 4,06$ horas. En la tabla 5.6 se observa la comparación de las

5.2.1 Resultados circuito sur

Como resultado de la aplicación del modelo matemático desarrollado a la data del Circuito Sur, se mejora notablemente el tiempo de auditoría obtenido durante la solución inicial (mejora de 9,13%). Las rutas obtenidas bajo el modelo son diferentes a la solución inicial, hasta el punto en que se requiere un ciclo menos para completar las auditorías del circuito.

Los ciclos obtenidos mediante el modelo son:

- **Ciclo #1:** Guayaquil - Machala - Milagro - Babahoyo - Ventanas - Quevedo - Vinces - Guayaquil.
- **Ciclo #2:** Guayaquil - Daule - Jipijapa - Portoviejo - B. Caráquez - Manta - Salinas - Villamil - Guayaquil.
- **Ciclo #3:** Vacío
- **Ciclo #4:** Guayaquil - Cuenca - Loja - Guayaquil.

Tabla 5.7 Solución Circuito Sur (Elaboración propia)

Ciclo	Ciudades visitadas	Locales visitados	Duración de Ciclos (min)	Duración de Ciclos (h)
1	6	10	2343	39.4
2	7	11	2366	39.1
3	-	-	-	-
4	2	2	1883	31.3
F.O.			6592 min	109.9 h

La respuesta obtenida mediante el modelo es el óptimo global, mejorando significativamente la respuesta obtenida por la solución inicial. En este caso, se puede comprobar la utilidad de las ecuaciones (21) y (22), gracias a las cuales se obtiene un ciclo menos que el estimado. El promedio de duración de ciclos es de $36,60 \pm 3,74$. En la Tabla 5.8 se puede observar la comparación de las respuestas obtenidas en el programa LINGO. En esta instancia, se puede apreciar que la formulación GA ofrece tiempos computacionales significativamente menores a las otras formulaciones, bajo

CAPÍTULO 6:

Discusión

Debido a que el problema del vendedor viajero es un NP-duro en todas sus variantes, la inclusión de restricciones necesarias para garantizar que se cumplan las características específicas del problema aumenta la carga computacional durante la resolución del modelo, y a su vez, limita la capacidad de este para resolver instancias medianas y grandes por métodos exactos; esto suma importancia a la exploración de diferentes alternativas de restricciones de eliminación de subtours, en búsqueda de la reducción de tiempos computacionales, pues como se pudo observar en la resolución de los circuitos, se puede observar para el circuito sur que la mejora del tiempo computacional para con la formulación GA es significativamente mayor al resto.

Con el objetivo de evaluar el comportamiento del modelo, se compara los resultados de las 3 estrategias de eliminación de subtours consideradas, con el tiempo computacional obtenido de modelo básico del TSP para 3 instancias: Circuitos Norte y Sur del problema ($n=13$ y 3 ciclos estimados) y ($n=16$ y 4 ciclos estimados) respectivamente, y una instancia aleatoria creada de ($n=19$ y 6 ciclos estimados).

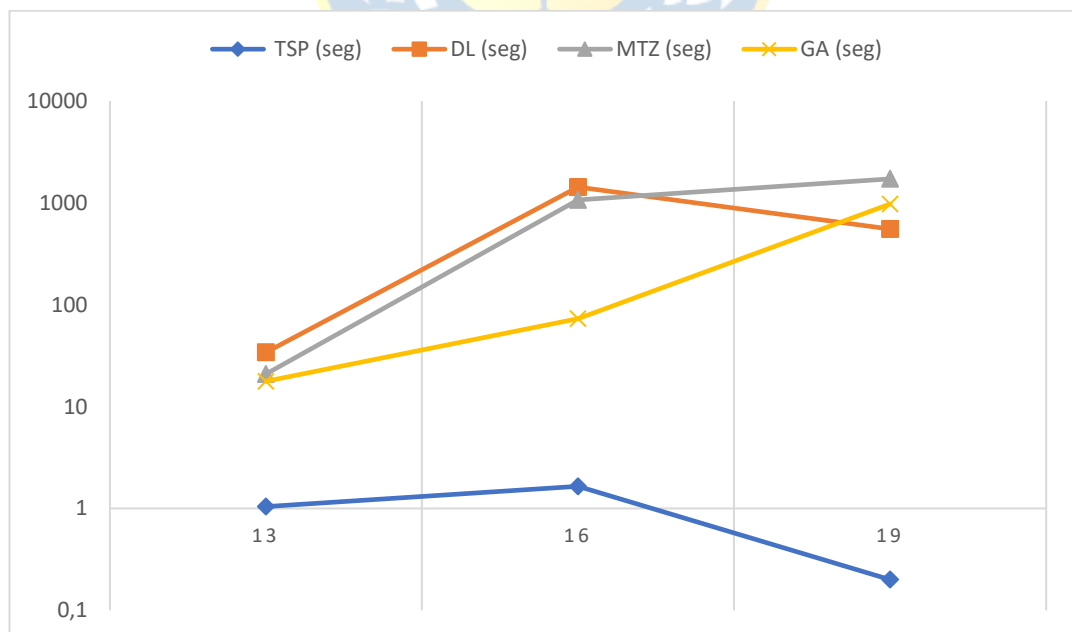


Figura 6.1 Tiempo computacional por instancia, por método de solución

Tabla 6.1 Tiempo computacional por instancia (Elaboración propia)

# Nodos	# Ciclos	TSP (s)	DL (s)	MTZ (s)	GA (s)
13	3	1.04	34.09	20.89	17.72
16	4	1.65	1428.84	1071.83	73.16
19	6	0.2	552.47	1720.56	974.78

Como se observa en la tabla 6.1, la formulación GA presenta un menor tiempo computacional en los circuitos generados para el problema, pero no en la instancia aleatoria; para la cual la formulación DL genera los mejores tiempos. Debe observarse también, que bajo la resolución del modelo TSP clásico, la instancia $n=19$ arrojó un menor tiempo computacional que las instancias del problema con menor número de nodos. Además de las tres instancias mencionadas, se corrió el modelo con una cuarta instancia ($n=21$), obtenida con los locales pertenecientes al Circuito Norte, sin ser agrupados por ciudades, no obteniendo respuesta para esta al cabo de 10 horas bajo ninguna formulación. De esta forma, se comprueba para el modelo la validez de las técnicas de clusterización para reducir el tiempo computacional de problemas de este tipo.

Para evaluar el impacto del número de ciclos en la repuesta del modelo desarrollado, se resuelve los Circuitos Norte y Sur clusterizados con valores fijos de $m=(3, 4, 5)$, y considerando únicamente la formulación DL, por haber arrojado los peores resultados en la resolución del modelo. Los ciclos (1 y 2) arrojan soluciones inviables para ambos circuitos, modificando las ecuaciones (15) y (16) de la siguiente forma:

$$\sum_{k=1}^m \sum_{i=1, i \neq j}^n X_{i1k} = m; \quad (22)$$

$$\sum_{k=1}^m \sum_{j=1, j \neq i}^n X_{1jk} = m; \quad (23)$$

Como es de esperar, se observa en la Tabla 6.2 que los tiempos computacionales de resolución aumentan con el número de ciclos, en parte porque con

ellos aumenta el número de variables de la respuesta. Además, se puede comprobar que la utilización de un número de ciclos fijo en la resolución del modelo, a pesar de ofrecer tiempos computacionales significativamente más cortos a los obtenidos en la resolución del problema del el Capítulo 5, no asegura la obtención del resultado óptimo, y hace necesario que el modelo explore un rango de ciclos que minimicen la solución, como se detalló en las ecuaciones (21) y (22).

Tabla 6.2 Respuesta del modelo al número de ciclos (Elaboración propia)

Circuito	# Nodos	# Ciclos	Respuesta	Incremento	Tiempo de solución (s)	Incremento
Norte	13	3	5515	-	3.83	
	13	4	5692	3.2%	5.67	48.0%
	13	5	5913	7.2%	5.55	44.9%
Sur	16	3	6592	-	25.99	-
	16	4	6632	0.6%	90.62	248.7%
	16	5	6709	1.8%	51.11	96.7%

Se explora también el impacto en el tiempo computacional de la amplitud de rango de ciclos que considera el modelo en su resolución, para esto, se resuelve el modelo desarrollado en ambos circuitos considerando tres escenarios de amplitud de búsqueda (A):

- A=2, obtenido de la solución del problema, con las ecuaciones (21) y (22), y con $m=3$ para el Circuito Norte, y $m=4$ para el Circuito Sur (valores obtenidos de la solución inicial).
- A=3, usando las ecuaciones (23) y (24) con $m=4$.
- A=5, usando las ecuaciones (25) y (26) con $m=5$.

Para obtener las amplitudes de búsqueda antes mencionadas, se modifica las ecuaciones (20) y (21) de la siguiente forma:

$$m - 2 \leq \sum_{k=1}^m \sum_{i=1, i \neq j}^n X_{i1k} \leq m; \quad (23)$$

$$m - 2 \leq \sum_{k=1}^m \sum_{j=1, j \neq i}^n X_{1jk} \leq m; \quad (24)$$

$$m - 4 \leq \sum_{k=1}^m \sum_{i=1, i \neq j}^n X_{i1k} \leq m; \quad (25)$$

$$m - 4 \leq \sum_{k=1}^m \sum_{j=1, j \neq i}^n X_{1jk} \leq m; \quad (26)$$

En la Tabla 6.3 se puede observar que independientemente de la amplitud de ciclo, siempre que los rangos explorados por el modelo incluyan el número de ciclos obtenidos de la solución inicial, se logra el óptimo; manteniendo también un aumento del tiempo computacional en función de la amplitud de rango de los ciclos.

Tabla 6.3 Respuesta del modelo a la amplitud de ciclo (Elaboración propia)

Circuito	# Nodos	# Ciclos	Amplitud	Respuesta	Tiempo de solución (s)	Incremento
Norte	13	2-3	2	5515	34.31	-
	13	2-4	3	5515	57.77	68.4%
	13	1-5	5	5515	3023.7	8712.9%
Sur	16	3-4	2	6592	1421.22	-
	16	2-4	3	6592	2213.31	55.7%
	16	1-5	5	6592	1763.92	24.1%

La dinámica del problema permite profundizar su estudio, con el desarrollo de otras variantes del TSP, tal como el MD-MTSP (*Multi Depot Multiple Traveling Salesman Problem*), y su comparación de sus resultados con el modelo desarrollado. De manera en que se estudie la conveniencia de visitar o no una ciudad que se encuentra en un circuito diferente, pero cercana a la posición del auditor en determinado momento de la jornada.

De igual forma, la incorporación de los costos de hora auditor, y de la posibilidad de que estos laboren tiempo extra, permitiría maximizar el uso de la jornada semanal de 40 horas, ya que para algunos de los ciclos generados en la solución del problema, se utilizaron tiempos menores a las 30 horas de la semana laboral.

Sin lugar a duda, la exploración del problema planteado debe continuarse mediante la inclusión de parámetros que incrementen la complejidad del sistema, y la incorporación de métodos de resolución metaheurísticos, que permitan obtener mejores tiempos de respuesta para el número de instancias que se requiere explorar en un problema de este tipo.



CAPÍTULO 7:

Conclusiones

- El problema SD-MTSP (Single Depot Multiple Salesman Problem) resultó adecuado para abordar la planificación de auditorías en una empresa de certificación, permitiendo bajo la modificación de sus restricciones clásicas, implementar una de las condiciones más importantes del servicio que es la limitación de jornadas laborales, bajo el costo computacional de incrementar el tamaño de la matriz de respuesta.
- El modelo desarrollado arrojó respuestas viables para su implementación en la planificación de jornadas de auditoría de la empresa en estudio, y abre la posibilidad de incluir locales de otras franquicias al modelo, para lograr una planificación global de sus actividades. Aunque la empresa sujeto de estudio, no cuenta con medidas de desempeño de los recorridos de sus auditores, las soluciones presentadas por el modelo representan una mejora significativa en las actividades de planificación de los servicios de auditoría para su cliente *Fast Food Inc.*
- El incremento en la carga computacional del modelo, además de ser influenciado por las restricciones de eliminación de subtours, que bajo las formulaciones exploradas demuestran variabilidad en el tiempo computacional de las soluciones, guarda relación con las restricciones que definen el potencial número de ciclos a explorar como parte de la respuesta. Es este contexto, al aproximar el número de ciclos requeridos para la solución del problema, mediante una heurística de tipo *NN*, y acotar entre $m-1$, y m ciclos la exploración de la respuesta, nos permite obtener respuestas del modelo en tiempos aceptables para las dos instancias probadas: $n=13$, $n=16$ y $n=19$.
- El uso de una determinada formulación específica para la eliminación de subtours no garantiza consistencia en su tiempo computacional para diferentes instancias, al compararse con otras formulaciones. Se pudo observar que la

formulación GA fue mejor en dos de las tres instancias probadas, mientras que la formulación DL fue mejor en la restante.

- Aunque el uso de técnicas de clusterización, y resolución del problema desde el enfoque del TSP generalizado, permite obtener mejores tiempos de respuesta para la solución exacta del modelo, limita junto con la capacidad combinatoria del mismo, la posibilidad de obtener mejores respuestas, ya que las respuestas obtenidas obligan a los auditores a visitar todos los locales de una ciudad como parte de una jornada, eliminando la posibilidad de que estos locales se visiten en múltiples ciclos. Para instancias de $n \geq 21$, no se obtuvieron tiempos computacionales en plazos razonables para el tamaño para las formulaciones utilizadas.



BIBLIOGRAFÍA

- Begur, S. V., Miller, D. M., & Weaver, J. R. (1997). An Integrated Spatial DSS for Scheduling and Routing Home-Health-Care Nurses. *Interfaces*, 27(4), 35-48. <https://doi.org/10.1287/inte.27.4.35>
- Bektas, T. (2006). The multiple traveling salesman problem: An overview of formulations and solution procedures. *Omega*, 34(3), 209-219. <https://doi.org/10.1016/j.omega.2004.10.004>
- Benhida, S., & Mir, A. (2018). Generating subtour elimination constraints for the Traveling Salesman Problem. *IOSR Journal of Engineering*, 08(7), 17-21.
- BRC Directory. (2020). <https://brcdirectory.co.uk/>
- Calviño, A. (2011). *Cooperación en los problemas del viajante (TSP) y de rutas de vehículos (VRP): Una panorámica*. Universidad de Santiago de Compostela.
- Chauhan, C., Gupta, R., & Pathak, K. (2012). Survey of Methods of Solving TSP along with its Implementation using Dynamic Programming Approach. *International Journal of Computer Applications*, 52(4), 12-19. <https://doi.org/10.5120/8189-1550>
- Cicková, Z., Brezina, I., & Pekar Juraj. (2008). Alternative Method for Solving Traveling Salesman Problem by Evolutionary Algorithm. *International Scientific Journal of Management Information Systems*, Vol. 3(1).
- Dodin, B., & Eliman, A. A. (1997). Audit scheduling with overlapping activities and sequence-dependent setup costs. *European Journal of Operational Research*, 97(1), 22-23.
- Gavish, B. (1976). Note—A Note on “The Formulation of the M-Salesman Traveling Salesman Problem”. *Management Science*, 22(6), 704-705. <https://doi.org/10.1287/mnsc.22.6.704>
- Gavish, B., & Srikanth, K. (1986). An Optimal Solution Method for Large-Scale Multiple Traveling Salesmen Problems. *Operations Research*, 34(5), 698-717. <https://doi.org/10.1287/opre.34.5.698>

- Hincapié, R., Ríos, C., & Gallego, R. (2004). Técnicas heurísticas aplicadas al problema del cartero viajante (tsp). *Scientia et Technica*, Vol.1(No. 24). <http://dx.doi.org/10.22517/23447214.7279>
- IFS Database. (2020). <https://www.ifs-certification.com/index.php/es/partners/certification-bodies>
- Kamarudin, A. A., Othman, Z. A., & Sarim, H. M. (2016). *Improvement Initial Solution Water Flow Like Algorithm Using Simulated Annealing for Travelling Salesman Problem*. 6, 4.
- Klansek, U. (2011). Using the TSP Solution for Optimal Route Scheduling in Construction Management. *Organization, technology & management in construction : an international journal*, Vol. 3(No. 1). <https://hrcak.srce.hr/69500>
- Kotthoff, L., Kerschke, P., Hoos, H., & Trautmann, H. (2015). Improving the State of the Art in Inexact TSP Solving Using Per-Instance Algorithm Selection. En C. Dhaenens, L. Jourdan, & M.-E. Marmion (Eds.), *Learning and Intelligent Optimization* (Vol. 8994, pp. 202-217). Springer International Publishing. https://doi.org/10.1007/978-3-319-19084-6_18
- Liu, S. (2014). A Powerful Genetic Algorithm for Traveling Salesman Problem. *ArXiv:1402.4699 [Cs]*. <http://arxiv.org/abs/1402.4699>
- Navab-Kashani, R., Gay, L. F., & Bayat, A. (2015). Productivity Improvement of Sewer CCTV Inspection through Time Study and Route Optimization. *Journal of Construction Engineering and Management*, 141(6), 04015009. [https://doi.org/10.1061/\(ASCE\)CO.1943-7862.0000976](https://doi.org/10.1061/(ASCE)CO.1943-7862.0000976)
- Öncan, T., Altinel, İ. K., & Laporte, G. (2009). A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, 36(3), 637-654. <https://doi.org/10.1016/j.cor.2007.11.008>

- Shuai, Y., Yunfeng, S., & Kai, Z. (2019). An effective method for solving multiple travelling salesman problem based on NSGA-II. *Systems Science & Control Engineering*, 7(2), 108-116. <https://doi.org/10.1080/21642583.2019.1674220>
- Stringer, M. (2005). Summary report. *Food Control*, 16(9), 775-794. <https://doi.org/10.1016/j.foodcont.2004.10.018>
- Svestka, J. A., & Huckfeldt, V. E. (1973). Computational Experience with an *M* - Salesman Traveling Salesman Algorithm. *Management Science*, 19(7), 790-799. <https://doi.org/10.1287/mnsc.19.7.790>
- Trienekens, J., & Zuurbier, P. (2008). Quality and safety standards in the food industry, developments and challenges. *International Journal of Production Economics*, 113(1), 107-122. <https://doi.org/10.1016/j.ijpe.2007.02.050>



ANEXOS

Anexo 1. Locales en Ecuador de la franquicia *Fast Food Inc.* (fuera de Quito y Guayaquil).

Provincia	Ciudad	# Local	Circuito	Duración (min)
Azuay	Cuenca	FF-19-Cuenca	Sur	120
Azuay	Cuenca	FF-30-Cuenca	Sur	180
Azuay	Cuenca	FF-59-Cuenca	Sur	120
Azuay	Cuenca	FF-102-Cuenca	Sur	120
Azuay	Cuenca	FF-134-Cuenca	Sur	120
El Oro	Machala	FF-49-Machala	Sur	180
El Oro	Machala	FF-50-Machala	Sur	120
El Oro	Machala	FF-104-Machala	Sur	180
Guayas	Daule	FF-36-Daule	Sur	120
Guayas	Milagro	FF-23-Milagro	Sur	120
Guayas	Villamil	FF-40-Villamil	Sur	120
Loja	Loja	FF-27-Loja	Sur	180
Loja	Loja	FF-79-Loja	Sur	120
Los Ríos	Babahoyo	FF-11-Babahoyo	Sur	120
Los Ríos	Babahoyo	FF-37-Babahoyo	Sur	180
Los Ríos	Quevedo	FF-52-Quevedo	Sur	180
Los Ríos	Quevedo	FF-61-Quevedo	Sur	120
Los Ríos	Ventanas	FF-105-Ventanas	Sur	180
Los Ríos	Vinces	FF-98-Vinces	Sur	120
Manabí	B. Caráquez	FF-73-B. Caráquez	Sur	120
Manabí	Jipijapa	FF-100-Jipijapa	Sur	180
Manabí	Manta	FF-90-Manta	Sur	120
Manabí	Manta	FF-95-Manta	Sur	120
Manabí	Manta	FF-99-Manta	Sur	120
Manabí	Manta	FF-126-Manta	Sur	180

Provincia	Ciudad	# Local	Circuito	Duración (min)
Manabí	Portoviejo	FF-129-Portoviejo	Sur	120
Manabí	Portoviejo	FF-135-Portoviejo	Sur	180
Santa Elena	Salinas	FF-141-Salinas	Sur	120
Carchi	Tulcán	FF-107-Tulcán	Norte	120
Chimborazo	Riobamba	FF-38-Riobamba	Norte	180
Chimborazo	Riobamba	FF-133-Riobamba	Norte	120
Cotopaxi	Latacunga	FF-20-Latacunga	Norte	120
Esmeraldas	Esmeraldas	FF-57-Esmeraldas	Norte	120
Esmeraldas	Atacames	FF-71-Atacames	Norte	180
Imbabura	Ibarra	FF-62-Ibarra	Norte	180
Imbabura	Ibarra	FF-69-Ibarra	Norte	120
Imbabura	Ibarra	FF-113-Ibarra	Norte	180
Orellana	El Coca	FF-106-El Coca	Norte	180
Pichincha	Cayambe	FF-68-Cayambe	Norte	180
Sto.	Sto. Domingo	FF-39-Sto. Domingo	Norte	120
Sto.	Sto. Domingo	FF-83-Sto. Domingo	Norte	120
Sto.	Sto. Domingo	FF-121-Sto. Domingo	Norte	180
Sto.	Sto. Domingo	FF-123-Sa Domingo	Norte	180
Sucumbíos	Lago Agrio	FF-74-Lago Agrio	Norte	180
Tungurahua	Ambato	FF-24-Ambato	Norte	120
Tungurahua	Ambato	FF-67-Ambato	Norte	120
Tungurahua	Ambato	FF-77-Ambato	Norte	180

Anexo 2. Matriz de distancias y duración de auditorías para el Circuito Norte (minutos)

	Quito (origen)	Tulcán	Riobamba	Latacunga	Esmeraldas	Atacames	Ibarra	El Coca	Cayambe	Sto. Domingo	Lago Agrío	Ambato	Baños
Quito (origen)	0	271	212	121	330	350	143	307	97	200	344	155	198
Tulcán	271	0	441	355	348	379	142	307	205	432	249	391	435
Riobamba	212	441	0	111	433	472	306	388	262	288	499	72	87
Latacunga	121	355	111	0	343	362	225	374	177	197	410	55	103
Esmeraldas	330	348	433	343	0	46	274	605	363	169	637	386	429
Atacames	350	379	472	362	46	0	303	624	361	188	655	402	447
Ibarra	143	142	306	225	274	303	0	386	63	295	326	259	303
El Coca	307	307	388	374	605	624	386	0	330	454	140	355	307
Cayambe	97	205	262	177	363	361	63	330	0	245	377	214	259
Sto. Domingo	200	432	288	197	169	188	295	454	245	0	488	229	272
Lago Agrío	344	249	499	410	637	655	326	140	377	488	0	435	404
Ambato	155	391	72	55	386	402	259	355	214	229	435	0	61
Baños	198	435	87	103	429	447	303	307	259	272	404	61	0
Duración	0	120	320	120	120	180	520	180	180	660	180	460	180

Anexo 3. Matriz de distancias y duración de auditorías para el Circuito Sur (minutos)

	Guayaquil	Cuenca	Machala	Daule	Milagro	Villamil	Loja	Babahoyo	Quevedo	Ventanas	Vinces	B. Caráquez	Jipijapa	Manta	Portoviejo	Salinas
Guayaquil	0	199	182	58	51	82	398	78	179	122	108	259	148	201	201	189
Cuenca	199	0	189	242	185	283	226	226	320	268	273	447	338	387	385	328
Machala	182	189	0	237	180	278	266	280	314	263	268	441	332	381	379	322
Daule	58	242	237	0	85	178	448	86	137	120	57	219	110	162	160	165
Milagro	51	185	180	85	0	141	386	52	159	104	103	294	188	238	235	186
Villamil	82	283	278	178	141	0	484	160	251	204	170	315	208	258	256	92
Loja	398	226	266	448	386	484	0	423	518	505	510	633	524	573	569	519
Babahoyo	78	226	280	86	52	160	423	0	120	66	63	295	187	238	235	211
Quevedo	179	320	314	137	159	251	518	120	0	67	108	207	224	215	171	294
Ventanas	122	268	263	120	104	204	505	66	67	0	70	244	224	247	205	259
Vinces	108	273	268	57	103	170	510	63	108	70	0	256	166	216	185	217
B. Caráquez	259	447	441	219	294	315	633	295	207	244	256	0	113	83	73	298
Jipijapa	148	338	332	110	188	208	524	187	224	224	166	113	0	56	53	194
Manta	201	387	381	162	238	258	573	238	215	247	216	83	56	0	49	235
Portoviejo	201	385	379	160	235	256	569	235	171	205	185	73	53	49	0	242
Salinas	189	328	322	165	186	92	519	211	294	259	217	298	194	235	242	0
Duración	0	740	520	120	120	120	320	320	320	180	120	120	180	600	320	120

Anexo 4. Modelación en LINGO del problema SD-MTSP desarrollado.

MODEL:

```
SETS:
    LOCAL / 1... N/: U, T;
    CICLO /1... M/;
    RUTA(LOCAL,LOCAL) :DIST;
    ASIGNACION(CICLO,LOCAL,LOCAL) :X;
ENDSETS

DATA:
    DIST = @OLE(C:\Users\Dell\Desktop\RESPUESTA Circuito
    Sur.xlsx);
    T = @OLE(C:\Users\Dell\Desktop\RESPUESTA Circuito
    Sur.xlsx);
    @OLE(C:\Users\Dell\Desktop\RESPUESTA Circuito
    Sur.xlsx)=X;
ENDDATA

N = @SIZE(LOCAL);
M = @SIZE(CICLO);

!Función Objetivo, minimización de la distancia
recorrida entre locales, por el tiempo de auditoría en
cada ciclo;
MIN = @SUM(ASIGNACION:(DIST*X)) + @SUM(ASIGNACION:(T*X));

!Para todos los ciclos, excepto el origen, cada nodo
recibe un arco, y un arco parte de cada nodo;
@FOR(LOCAL(K) | K#GT#1:
    @SUM(CICLO(L) : @SUM(LOCAL(I) | I#NE#K:X(L,I,K))) = 1;
    @SUM(CICLO(L) : @SUM(LOCAL(J) | J#NE#K:X(L,K,J))) = 1;
);

@FOR(CICLO(L) :
    !Por cada ciclo, no más de un arco parte del origen,
    y retorna al mismo;
    @FOR(LOCAL(K) | K#EQ#1 : @SUM(LOCAL(I) | I#NE#K:X(L,I,K)) <
=1);
    @FOR(LOCAL(K) | K#EQ#1 : @SUM(LOCAL(J) | J#NE#K:X(L,K,J)) <
=1);
);
```

```

!Un arco parte del origen, y retorna al mismo entre M-1
y M veces, siendo M el número de ciclos estimados en la
solución inicial;
@FOR (LOCAL (K) | K#EQ#1:
    @SUM (CICLO (L) : @SUM (LOCAL (I) | I#NE#K: X (L, I, K) ) ) <= M;
    @SUM (CICLO (L) : @SUM (LOCAL (I) | I#NE#K: X (L, I, K) ) ) >= M-1;
    @SUM (CICLO (L) : @SUM (LOCAL (J) | J#NE#K: X (L, K, J) ) ) <= M;
    @SUM (CICLO (L) : @SUM (LOCAL (J) | J#NE#K: X (L, K, J) ) ) >= M-1;
);

@FOR (CICLO (L) :
    !Cada ciclo tiene una duración máxima de 40
    horas(2400 min);
    @SUM (ASIGNACION (L, I, J) | I#NE#J: ( (DIST*X) + (T*X) ) ) <=240
    0;

    !Restricciones de eliminación de subtours;
    !ALTERNATIVA 1: Formulación DL (Desrochers &
    Laporte);
    @FOR (LOCAL (K) : @FOR (LOCAL (J) | J#GT#1 #AND#
    J#NE#K: U (K) -U (J) + (N-1) *X (L, K, J) + (N-3) *X (L, J, K) <=N-2 ) );
    !ALTERNATIVA 2: Formulación MTZ (Miller-Tucker-
    Zemlin);
    @FOR (LOCAL (K) : @FOR (LOCAL (J) | J#GT#1 #AND#
    J#NE#K: U (K) -U (J) +N*X (L, K, J) <=N-1 ) );
    !ALTERNATIVA 3: Formulación G (Gavish);
    @FOR (LOCAL (K) : @FOR (LOCAL (J) | J#GT#1 #AND#
    J#NE#K: U (K) -U (J) + (N-M) *X (L, K, J) <=N-M-1 ) );

    !Restricción de continuidad dentro del ciclo (de
    cada nodo que recibe un arco, parte otro);
    @FOR (LOCAL (K) : @SUM (LOCAL (J) | J#NE#K: X (L, K, J) ) -
    @SUM (LOCAL (I) | I#NE#K: X (L, I, K) ) =0 ) ;
);

@FOR (ASIGNACION: @BIN (X) );

END

```

Anexo 5. Modelación en LINGO del problema TSP.

MODEL:

```
SETS:
    LOCAL / 1... N/: U, T;
    RUTA (LOCAL, LOCAL) :Y, DIST;
ENDSETS

DATA:
    DIST = @OLE(C:\Users\Dell\Desktop\RESPUESTA Circuito
    Sur.xlsx);
    T = @OLE(C:\Users\Dell\Desktop\RESPUESTA Circuito
    Sur.xlsx);
    @OLE(C:\Users\Dell\Desktop\RESPUESTA Circuito
    Sur.xlsx)=Y;
ENDDATA

N = @SIZE(LOCAL);

! Función Objetivo, minimización de la distancia
recorrida entre locales;
MIN = @SUM(RUTA: (DIST*Y)) + @SUM(RUTA: (T*Y));

! Cada nodo recibe un arco, y un arco parte de cada
nodo;
@FOR(LOCAL(K) :
    @SUM(LOCAL(I) | I#NE#K: Y(I, K)) = 1;
    @SUM(LOCAL(J) | J#NE#K: Y(K, J)) = 1;

! Formulación DL (Desrochers & Laporte) para la
eliminación de subtours en cada ciclo;
@FOR(LOCAL(J) | J#GT#1 #AND# J#NE#K:
    U(K) - U(J) + (N-1) * Y(K, J) + (N-3) * Y(J, K) <= N-2;
);

@FOR(RUTA: @BIN(Y));

END
```