



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

FACULTAD DE CONTADURÍA, ADMINISTRACIÓN E INFORMÁTICA
MAESTRÍA EN OPTIMIZACIÓN Y CÓMPUTO APLICADO

**Aplicación del modelo WCVRPTW al problema de
recolección de residuos sólidos de empresas del
sector turístico**

T E S I S

Que para obtener el Grado de MAESTRO EN OPTIMIZACIÓN Y
COMPUTO APLICADO

Presenta

ING. DIEGO GUSTAVO HURTADO OLIVARES

Director de Tesis

DR. JOSÉ ALBERTO HERNÁNDEZ AGUILAR

Co-Director

DR. JOSÉ CRISPÍN ZAVALA DÍAZ

Revisores:

DR. FEDERICO ALONSO PECINA, DR. CARLOS ALBERTO OCHOA
ORTIZ, DR. MARTÍN HERIBERTO CRUZ ROSALES



CUERNAVACA, MORELOS

NOVIEMBRE, 2020

Dedictory

This thesis is dedicated to my parents for their love, endless support and encouragement and to God Almighty my source of inspiration, wisdom, knowledge and understanding.

A special feeling of gratitude to my loving parents, Sandra and Gustavo for all their love and encouragement. Who raised me with love and supported me in all my goals. My sisters Ashley and Fernanda have never left my side and are very special.

I also dedicate this work to my girlfriend; Mafer who has encouraged me all the way and most of all for my loving, supportive, encouraging, and patient girlfriend whose faithful support during the final stages of this MSC is so appreciated. Thank you

I am grateful for time spent with friends, for our memorable memories.

Dr. Alberto Hernández who provided me the encouragement, support and inspiration that made this project possible.

Acknowledges

Thank you first and foremost to my thesis director, Dr. Alberto Hernández, for giving me the opportunity to do research in this unique field; I appreciate all his contributions of time, ideas and to make my MSC experience productive and stimulating. The joy and enthusiasm he has for his research was contagious and motivational for me, even during tough times in the MSC pursuit. Thank you to my research advisor, Dr. Crispín Zavala, for his guidance throughout this process, and for introducing me to much of the methodology that underlies this work.

I would be remiss without mentioning some professors who have had a particular influence on my work, and on my appreciation of optimization, Dr. Federico Pecina.

Dr. Alberto Ochoa his ongoing feedback, support, and guidance as well as consistent pushing of my thinking challenged me in all the right ways.

I would also like to thank my committee members provided amazing feedback and support during this process.

Resumen

Esta investigación presenta la utilización de procedimientos de optimización combinatoria para resolver el problema de la recolección de desechos sólidos en el sector turístico. La recolección de residuos es una actividad importante de logística dentro de las ciudades. Esta problemática se resuelve como el problema de enrutamiento de vehículos para optimizar la recolección de residuos sólidos, y se extiende al problema con capacidad limitada con ventanas de tiempo para aumentar la eficiencia y minimizar el costo total de la ruta de recolección. Esta tesis aborda el VRPTW mediante el uso del algoritmo de recocido simulado con métodos de vecindad para mejorar la solución. Se desarrollan funciones especiales para explorar eficazmente el espacio de búsqueda mediante métodos de vecindad. El enfoque propuesto se evalúa en comparación con las instancias del problema del VRPTW de referencia disponibles en la literatura. Un conjunto de experimentos computacionales demuestra que el recocido simulado mejorado podría producir resultados prometedores en el tiempo computacional de 180 segundos promedio. Los resultados relacionados indican que el algoritmo puede reducir significativamente el costo a un 12.27 % del óptimo global en el conjunto de instancias de prueba RC. Además, nuestra propuesta se emplea para un caso de estudio del mundo real para la recolección de residuos sólidos y con ello se puede comprobar la efectividad de optimizar las rutas de recolección para la instancia real de prueba. Debido a la rica naturaleza del problema, los solucionadores comerciales de vanguardia solo pueden abordar instancias de tamaño pequeño a mediano. Para resolver casos reales, proponemos un recocido simulado capaz de tratar sistemáticamente todas las características del problema y lo suficientemente general como para responder a las diferentes características de las regiones del caso de estudio para el cual está destinado. Los resultados muestran que la heurística logra la optimización de la instancia, exhibe un rendimiento competitivo en comparación con métodos de solución de vanguardia para casos especiales de nuestro problema, y con ello se pretende conducir a importantes ahorros en el estado de práctica.

Abstract

This research presents an application of combinatorial optimization techniques to address solid waste collection in the tourism sector. The waste collection is an important logistics activity within cities. This problem is modeled as the vehicle routing problem to improve the efficiency of solid waste collection and extends to the problem with limited capacity with time windows to increase efficiency and minimize the total cost of the collection route. This thesis addresses the VRPTW by using the simulated annealing algorithm with neighborhood methods to improve the solution. Special neighborhood methods are developed to effectively explore the space search. The proposed approach is evaluated and compared to the VRPTW benchmark instances available in the literature. A set of computational experiments shows that improved simulated annealing could produce promising results in the computational time average of 180 seconds. The related results indicate that the algorithm can significantly reduce the cost to 12.27% of the global optimum in the set of test instances RC. In addition, our proposal is used for a real-world case study of a solid waste collection and with this we were able to verify the effectiveness of optimizing the collection routes for the actual test instance. Due to the rich nature of the problem, leading edge business solvers can only address small to medium-sized instances. To solve real cases, we propose a simulated annealing capable of systematically treating all the characteristics of the problem and general enough to respond to the different characteristics of the case study regions for which it is intended. The results show that heuristics achieve in-instance optimization, exhibit competitive performance compared to leading-edge solution methods for special cases of our problem, and it is intended to lead to significant savings in practice.

Tabla de contenido

DEDICATORY	II
ACKNOWLEDGES	III
RESUMEN	IV
ABSTRACT	V
TABLA DE CONTENIDO	VI
LISTA DE FIGURAS	VIII
LISTA DE TABLAS	IX
<u>CAPÍTULO 1. INTRODUCCIÓN</u>	<u>1</u>
1.1. ANTECEDENTES	1
1.2. PANORAMA GENERAL DEL PROYECTO	2
1.3. OBJETIVO GENERAL	3
1.4. OBJETIVOS ESPECÍFICOS	3
1.5. JUSTIFICACIÓN	3
1.6. ALCANCES Y LIMITACIONES	4
1.7. HIPÓTESIS	4
1.8. IMPACTO	4
1.9. MATRIZ DE CONGRUENCIA	5
1.10. ORGANIZACIÓN DE LA TESIS	7
<u>CAPÍTULO 2. MARCO TEÓRICO DEL PROBLEMA DE ENRUTAMIENTO DE VEHÍCULOS</u>	<u>8</u>
2.1. SMART CITIES	8
2.2. TRANSPORTACIÓN INTELIGENTE	9
2.3. RECOLECCIÓN DE RESIDUOS	11
2.3.1. CLASIFICACIÓN DE RESIDUOS	14
2.3.2. ASPECTOS RELACIONADOS CON LOS COSTOS MONETARIOS	14
2.3.3. ASPECTOS AMBIENTALES: CONTAMINACIÓN DEL AIRE Y SUELO	16
2.4. ENRUTAMIENTO DE VEHÍCULOS ENFOCADO A LA RECOLECCIÓN DE RESIDUOS SÓLIDOS	17
2.5. TEORÍA DE GRAFOS	20
2.5.1. MATRIZ DE ADYACENCIA	22
2.6. OPTIMIZACIÓN COMBINATORIA	23
2.7. COMPLEJIDAD	23
2.8. MÉTODOS EXACTOS	25
2.9. HEURÍSTICAS CLÁSICAS	26
2.10. METAHEURÍSTICA	27
2.11. ORIGEN DEL VRP	27
2.11.1. PROBLEMA DEL AGENTE VIAJERO (TSP)	30
2.11.2. EL PROBLEMA DEL MÚLTIPLE AGENTE VIAJERO (MTSP)	32
2.11.3. EL PROBLEMA DE ENRUTAMIENTO DE VEHÍCULOS (VRP)	36
2.12. VARIANTES DEL VRP	41
2.12.1. VRPPD	43
2.12.2. VRPB	43
2.12.3. MDVRP	43
2.12.4. OVRP	44

2.12.5. DVRP	44
2.12.6. VRPTW	45
2.12.7. WCVRPTW	49
2.13. TABLA DE RESTRICCIONES DE LOS TIPOS DE PROBLEMAS DE RUTEO	52
2.14. DIFERENTES MÉTODOS DE RESOLUCIÓN DEL VRPTW	53
2.15. TRABAJO RELACIONADO DE WCVRPTW	56
2.15.1. ALGORITMOS UTILIZADOS PARA RESOLVER EL WCVRPTW	64
<u>CAPÍTULO 3. METODOLOGÍA PARA LA RESOLUCIÓN DEL PROBLEMA</u>	66
3.1. PLANTEAMIENTO	66
3.1.1. DEFINICIÓN DEL MODELO MATEMÁTICO	67
3.1.2. NOTACIÓN MATEMÁTICA	69
3.1.3. PARÁMETROS DE LOS OBJETOS	72
3.1.4. VARIABLES DE DECISIÓN	74
3.1.5. RESTRICCIONES	75
3.1.6. MODELO MATEMÁTICO	75
3.2. DESCRIPCIÓN DEL ALGORITMO PARA RESOLVER EL VRPTW	79
3.2.1. REPRESENTACIÓN DE LA SOLUCIÓN	79
3.2.2. SOLUCIÓN INICIAL	80
3.2.3. OPERADORES DE MEJORA DE LA SOLUCIÓN	83
3.2.3.1. 2-OPT	86
3.2.3.2. INTERCAMBIO	88
3.2.3.3. RELOCALIZACIÓN	89
3.2.4. RECOCIDO SIMULADO	90
3.2.4.1. PARÁMETROS	94
3.2.4.2. PERTURBACIÓN DE LA SOLUCIÓN	96
3.2.4.3. FUNCIÓN DE ACEPTACIÓN	97
3.3. DISEÑO EXPERIMENTAL	98
3.4. INSTANCIAS VRPTW	102
3.5. CASO DE ESTUDIO	108
3.5.1. EXTRACCIÓN DE DATOS	113
3.5.2. MATRIZ DE COSTOS	115
3.5.2.1. PARÁMETROS OPCIONALES	119
3.5.3. VENTANAS DE TIEMPO DE CLIENTES	120
3.5.4. VISUALIZACIÓN DE RUTAS	122
3.5.5. OPTIMIZACIÓN DE RUTAS MEDIANTE API	123
<u>CAPÍTULO 4. RESULTADOS Y DISCUSIÓN</u>	125
4.1. RESULTADOS VRPTW	125
4.2. RESULTADOS CASO DE ESTUDIO: VRPTW	128
<u>CAPÍTULO 5. CONCLUSIONES Y TRABAJO FUTURO</u>	137
5.1. CONCLUSIONES	137
5.1. TRABAJO FUTURO	139
<u>REFERENCIAS BIBLIOGRÁFICAS</u>	143

Lista de figuras

FIGURA 2.1. ESQUEMA DE SMART CITIES (TECH DATA CORPORATION, 2020)	9
FIGURA 2.2. PROBLEMAS DE TRANSPORTE (FUENTE PROPIA).	10
FIGURA 2.3. DISEÑO DE RUTAS DE RECOLECCIÓN EN MÉXICO, FUENTE (INAPAM, 2017)	13
FIGURA 2.4. CONTAMINACIÓN EN CIUDAD DE MÉXICO (LÓPEZ DORIGA, 2015)	17
FIGURA 2.5. REPRESENTACIÓN DE RED DE CLIENTES, USUALMENTE SUELE ESTAR PRESENTADA POR UN GRAFO (GOOGLE DEVELOPERS, 2020).....	18
FIGURA 2.6. EVOLUCIÓN DEL PROBLEMA DE WCVRPTW.....	19
FIGURA 2.7. CICLO HAMILTONIANO.....	21
FIGURA 2.8. MATRIZ DE ADYACENCIA	22
FIGURA 2.9. COMPLEJIDAD DEL VRPTW SIMÉTRICO (LENSTRA & KAN, 1981).....	24
FIGURA 2.10. COMPLEJIDAD DEL VRPTW NO SIMÉTRICO (LENSTRA & KAN, 1981)	25
FIGURA 2.11. ILUSTRACIÓN DEL PROBLEMA DE AGENTE VIAJERO (SCILAB, 2020)	30
FIGURA 2.12. DIFERENCIA ENTRE TSP Y MTSP (FUENTE PROPIA).....	32
FIGURA 2.13. EJEMPLO DE UNA SOLUCIÓN DEL PROBLEMA DE ENRUTAMIENTO DE VEHÍCULOS CON 1 DEPÓSITO, 25 CLIENTES Y 5 VEHÍCULO (LIM & WANG, 2005)	36
FIGURA 2.14. COORDENADAS DE CLIENTES (GOOGLE DEVELOPERS, 2020).....	38
FIGURA 2.15. ASIGNACIÓN DE RECORRIDOS ÓPTIMOS (COLOR) (FUENTE PROPIA)	38
FIGURA 2.16. CODIFICACIÓN DE ASIGNACIÓN DE RUTAS A VEHÍCULOS.....	39
FIGURA 2.17. VARIANTES VRP BASADO EN (IRNICH, SCHNEIDER & VIGO, 2014).....	42
FIGURA 2.18. VENTANA DE TIEMPO DE VEHÍCULO	48
FIGURA 2.19. ESQUEMA DE WCVRPTW (FUENTE PROPIA).....	50
FIGURA 2.20. WCVRPTW EN UNA SMART CITY (FUENTE PROPIA)	51
FIGURA 2.21. MÉTODOS PARA RESOLVER EL VRPTW (FUENTE PROPIA).....	54
FIGURA 2.22. ALGORITMOS PARA RESOLVER EL WCVRPTW (FUENTE PROPIA)	65
FIGURA 3.1. REPRESENTACIÓN DE LA ASIGNACIÓN DE CLIENTES A LOS VEHÍCULOS	80
FIGURA 3.2. CRITERIOS DE ASIGNACIÓN DE CLIENTES A VEHÍCULOS.	81
FIGURA 3.3. ASIGNACIÓN DE LOS CLIENTES AL K VEHÍCULO.....	82
FIGURA 3.4. ILUSTRACIÓN DE LA VECINDAD 2-OPT.	87
FIGURA 3.5. ILUSTRACIÓN DE LA VECINDAD DE INTERCAMBIO	88
FIGURA 3.6. ILUSTRACIÓN DE LA VECINDAD DE RELOCALIZACIÓN	90
FIGURA 3.7. ALGORITMO DE RECOCIDO SIMULADO CON OPERADORES DE VECINDAD.....	93
FIGURA 3.8. PARÁMETROS DE RECOCIDO SIMULADO BASADOS EN (MAHMUDY, 2014)	95
FIGURA 3.9. MÉTODOS DE VECINDAD PARA RECOCIDO SIMULADO	97
FIGURA 3.10. PROCESO EXPERIMENTAL McGEOCH, (2012).....	99
FIGURA 3.11. PASOS PARA LA RESOLUCIÓN DE LAS INSTANCIAS PROPUESTAS.....	100
FIGURA 3.12. CLASIFICACIÓN DE LAS INSTANCIAS DE SOLOMON PARA EL VRPTW (SOLOMON, 1987). ...	102
FIGURA 3.13. SITIOS DE RECOLECCIÓN DE RESIDUOS SÓLIDOS CUERNAVACA, MORELOS.	110
FIGURA 3.14. METODOLOGÍA PARA RESOLVER INSTANCIA REAL	111
FIGURA 3.15. DIFERENCIAS DE DISTANCIAS DE INSTANCIAS E INSTANCIAS REALES.....	112
FIGURA 3.16. SITIOS DE RECOLECCIÓN DE RESIDUOS SÓLIDOS CUERNAVACA, MORELOS.	115
FIGURA 3.17. CONVERSIÓN DE HORARIOS DE ATENCIÓN.....	121
FIGURA 3.18. EJEMPLO DE VENTANA DE TIEMPO CLIENTE	121
FIGURA 3.19. REPRESENTACIÓN DE DIRECCIONES PARA RUTA.....	122
FIGURA 3.20. SITIOS DE RECOLECCIÓN DE RESIDUOS SÓLIDOS CON DIRECCIONES	123
FIGURA 3.21. COMPARACIÓN DE INSTANCIAS, API Y SOLUCIÓN A LA MEDIDA.....	124
FIGURA 4.1. CONJUNTO DE CLIENTES A VISITAR	129
FIGURA 4.2. CONSIDERACIONES PARA RESOLVER INSTANCIAS REALES	130
FIGURA 4.3. VISUALIZACIÓN DE RUTA A) OPTIMIZADA CON ALGORITMO PROPUESTO	131
FIGURA 4.4. VISUALIZACIÓN DE ASIGNACIÓN DE RUTA A) CON API DE GOOGLE.....	132
FIGURA 4.5. VISUALIZACIÓN DE ASIGNACIÓN DE SEGUNDA RUTA B) CON ALGORITMO PROPUESTO	133
FIGURA 4.6. VISUALIZACIÓN DE ASIGNACIÓN DE SEGUNDA RUTA B) CON API DE GOOGLE.....	134

FIGURA 4.7. VISUALIZACIÓN DE ASIGNACIÓN DE RUTAS CON A) ALGORITMO PROPUESTO Y B) API DE GOOGLE.....	135
FIGURA 5.1. CRONOGRAMA DE ACTIVIDADES DEL PROBLEMA DE INVESTIGACIÓN	138
FIGURA 5.2. FASES PARA REALIZADAS COMPLETAR EL PROBLEMA DE INVESTIGACIÓN	139
FIGURA 5.3. PROPUESTAS DE UN DISEÑO INTELIGENTE DE RECOLECCIÓN DE BASURA	140
FIGURA 5.4. OPTIMIZACIÓN DE RUTAS TOMANDO DATOS ESTADÍSTICOS DE VIAJES.....	142

Lista de tablas

TABLA 1.1. TABLA MATRIZ DE CONGRUENCIA.....	5
TABLA 2.1. PROCEDIMIENTOS DE SOLUCIÓN PROPUESTOS PARA EL MTSP.	34
TABLA 2.2. TABLA DE RESTRICCIONES DE LOS MODELOS DE ENRUTAMIENTO DE VEHÍCULOS.....	52
TABLA 2.3. TRABAJO RELACIONADO AL RUTEO VEHICULAR	62
TABLA 2.4. PROCEDIMIENTOS DE SOLUCIÓN PROPUESTOS PARA EL WCVRPTW.	65
TABLA 3.1. NOTACIÓN DE OBJETOS	69
TABLA 3.2. TABLA NOTACIÓN DEL MODELO MATEMÁTICO	72
TABLA 3.3. TABLAS VARIABLES DE DECISIÓN DEL MODELO MATEMÁTICO	74
TABLA 3.4. TABLA DE MÉTODOS DE MEJORAR PARA LA SOLUCIÓN DEL VRPTW	84
TABLA 3.5. TABLA DE PRINCIPALES CARACTERÍSTICAS DE LA HEURÍSTICA PARA VRPTW	84
TABLA 3.6. TABLA DE PARÁMETROS DE RECODIDO SIMULADO	95
TABLA 3.7. CARACTERÍSTICA PRINCIPAL DE LAS INSTANCIAS DEL VRPTW (SOLOMON, 1987).	103
TABLA 3.8. ESTRUCTURAS DE LAS INSTANCIAS DEL VRPTW (SOLOMON, 1987).	103
TABLA 3.9. INSTANCIA C101 DEL CONJUNTO DE INSTANCIAS DE SOLOMON (SOLOMON-BENCHMARK, 2010)	104
TABLA 3.10. TABLA DE MEJORES RESULTADOS DE LAS INSTANCIAS DEL VRPTW (SOLOMON-BENCHMARK, 2010).....	105
TABLA 3.11. TABLA DE AUTORES QUE TUVIERON LOS MEJORES RESULTADOS Y MÉTODO QUE UTILIZARON (SOLOMON-BENCHMARK, 2010).....	106
TABLA 3.12. TABLA COMPARATIVA DE INSTANCIAS LITERATURA E INSTANCIAS REALES	112
TABLA 3.13. ESTRUCTURAS DE LAS INSTANCIAS REALES	113
TABLA 3.14. EJEMPLO DE MATRIZ DE ADYACENCIA DE GOOGLE.	116
TABLA 3.15. TABLA DE MATRIZ DE ADYACENCIA DE CLIENTES.....	117
TABLA 3.16. TABLA DE MATRIZ DE TIEMPO DE CLIENTES.	118
TABLA 4.1. TABLA DE RESULTADOS INSTANCIAS TIPO C.....	126
TABLA 4.2. TABLA DE RESULTADOS INSTANCIAS TIPO R.....	126
TABLA 4.3. TABLA DE RESULTADOS INSTANCIAS TIPO RC	127
TABLA 4.4. TABLA DE RESULTADOS OBTENIDOS POR TIPO DE INSTANCIAS	128
TABLA 4.5. COMPARACIÓN DE RUTA OBTENIDA POR ALGORITMO Y POR API DE GOOGLE.....	135

Capítulo 1. INTRODUCCIÓN

En este capítulo se describe el Problema de Enrutamiento de Vehículos para Recolección de Basura con Ventanas de Tiempo, partiendo de la descripción del estado actual de la problemática de la asignación de rutas de recolección de residuos sólidos; así mismo se presenta el problema de investigación, los objetivos e hipótesis, así como los resultados deseados de la tesis. Finalmente, se describen las limitaciones y supuestos del estudio.

1.1. Antecedentes

El manejo de residuos tiene un papel sumamente importante dentro de la planeación del aseo urbano, con este plan la salud pública de los ciudadanos se conserva mediante la recolección de los desechos generados para transportarlos a los centros de disposición final de forma eficiente y con menor costo posible, debido a que esta etapa emplea un número considerable de recursos económicos del sector público y privado.

La problemática consiste en realizar la asignación de los vehículos para recolectar los desechos de los clientes dentro de un tiempo determinado minimizando el costo de viaje total. A esto comúnmente se le conoce como ruta, ésta se encarga de asignar el itinerario específico que deben realizar diariamente los vehículos de recolección en las zonas de la localidad, con el fin de realizar la recolección de la mejor forma posible de los residuos generados por los establecimientos de dicho sector. Así, un gran potencial para obtener ahorros económicos y ambientales sustanciales radica en optimizar el transporte. Los procesos de planificación, optimización y control de las actividades logísticas y de transporte en zonas urbanas del flujo de los consumidores a las instalaciones de reciclaje o eliminación, respectivamente. Por lo tanto, la recolección de

residuos es un componente importante en la planificación logística dentro de las ciudades.

Esta tesis analiza diferentes soluciones y métodos para la realización de rutas de recolección de residuos. Si bien el objetivo del problema de enrutamiento de vehículos clásico es minimizar el costo total de la ruta, se podrían considerar y analizar otros objetivos como la minimización de tiempo, la minimización de vehículos, y la minimización de recursos, entre otros; sin embargo, nosotros abordamos el problema clásico de minimizar la distancia total de viaje. Este documento intenta servir como una guía en cuanto a la revisión de literatura de investigación producida en el campo de WCVRPTW, como propuestas de solución a la problemática, métodos de solución y resultados obtenidos que le puedan servir como ayuda tanto para investigadores como a profesionistas que estén enfrentando este tipo de problemáticas. Finalmente, también se identifican otros temas de investigación para ser estudiados como desarrollos futuros que se pudieran integrar a la investigación y enriquecer como son métodos estadísticos del historial de viajes, predicción de tiempo de traslado, estimación de generación de residuos entre otros.

1.2. Panorama general del proyecto

A continuación, se presentan datos relevantes de un estudio realizado por el (INAPAM, 2017) el cual muestra un panorama general de la situación actual de la recolección de residuos en las ciudades de México, “El 60.97% de los residuos sólidos generados proceden de fuentes domiciliarias y el restante 39.02% de las industrias, comercios y otras fuentes. Sólo el 43% de las ciudades medias realizan un diseño a través de un método técnico para llevar a cabo la recolección”. En cuanto a Morelos las rutas de recolección de residuos sólidos en el sector turístico y de servicios son determinadas de forma empírica, es por ello que se desea saber si se pueden optimizar estas rutas.

1.3. Objetivo General

Con el propósito de diseñar un sistema adecuado de recolección, se pretende encontrar una ruta factible para el problema de recolección de residuos sólidos en el sector turístico en Cuernavaca, Morelos.

Con esto, el fin de la investigación es verificar la viabilidad del método propuesto para la recolección de residuos mediante la optimización de rutas en términos de la distancia total del traslado de los vehículos.

1.4. Objetivos Específicos

- Modelar matemáticamente el WCVRP con ventanas de tiempo
- Revisar la literatura en el estado del arte del WCVRP con ventanas de tiempo en el sector turístico
- Realizar experimentos computacionales para calibración de los parámetros utilizados en las estrategias implementadas.
- Realizar una comparación de los resultados al utilizar la heurística propuesta y los resultados obtenidos por heurísticas reportadas en la literatura.
- Simular el problema WCVRPTW de la recolección de residuos sólidos en el sector turístico de Cuernavaca, Morelos
- Verificar con un caso de estudio si la solución propuesta es funcional
- Documentar los análisis, resultados y conclusiones, en un artículo publicable en una revista internacional.

1.5. Justificación

Es necesario que los municipios cuenten estrategias que les permiten planear y programar sus acciones incorporando nuevas tecnologías. Las dependencias responsables de la limpieza pública requieren de técnicas modernas para la planeación

de la asignación de rutas de recolección que incluyen técnicas de inteligencia artificial y de capacitación para el personal.

1.6. Alcances y limitaciones

La principal contribución de esta tesis es utilizar técnicas de optimización combinatoria, procesamiento y análisis de datos, así como técnicas de visualización para realizar la mejora de la recolección de residuos sólidos en el sector turístico. A partir de este proyecto de investigación se podrá resolver el problema de ruteo vehicular con ventanas de tiempo. En cuanto a la viabilidad del proyecto, se cuenta con instancias de prueba e información real para llevar a cabo todas las simulaciones requeridas. Para el tiempo de realización del proyecto se creó un cronograma de actividades para cada semestre con el cual se lleva la asignación de las tareas para realizar la culminación de éste. Sin embargo, para la realización del método de solución, solo se desarrollará una heurística. En casos reales, la planificación de rutas tiene mucha incertidumbre, como cambios en la demanda, accidentes de tráfico o condiciones climáticas inesperadas, etc. Todos estos factores no se toman en cuenta. A su vez no se tomará en cuenta en nuestro modelo el flujo vehicular sin embargo sí se realizará su visualización.

1.7. Hipótesis

H1. A través del uso de WCVRP con ventanas de tiempo se puede resolver el problema de recolección de basura en el sector turístico en Cuernavaca; Morelos

H0. A través del uso de WCVRP con ventanas de tiempo no se puede resolver el problema de recolección de basura en el sector turístico en Cuernavaca, Morelos.

1.8. Impacto

Se pretende minimizar la distancia total de viaje al diseñar rutas de menor costo para la recolección de residuos sólidos de clientes en diferentes ubicaciones, dada una flota de vehículos y un conjunto de restricciones. Una mejor planificación para la optimización

de ruta con el fin de que un camión pueda recolectar la mayor cantidad de residuos, la mayor cantidad de los clientes, en el menor tiempo, con la menor cantidad de kilómetros recorridos y con ello una ejecución más rápida. Hay enormes beneficios que brinda una solución con técnicas de inteligencia artificial como ahorrar tiempo, ahorrar dinero, incrementar la productividad, incrementar los márgenes de ganancias, aumentar la satisfacción de los clientes, la sostenibilidad, etc.

En este sentido se sabe que el crecimiento incontrolado, la urbanización rápida de las ciudades de México y las actividades diarias producen una gran cantidad de residuos en viviendas, áreas comerciales, o industriales en todo el mundo, y que estas actividades impactan en el clima al aumentar la emisión de diferentes gases de efecto invernadero. Por lo que la calidad del medio ambiente se está deteriorando rápidamente con el consiguiente daño en la salud de los mexicanos.

1.9. Matriz de congruencia

La matriz de congruencia es una estrategia metodológica valiosa que nos permite diseñar de forma general el proceso investigación que vamos a realizar. Nos asegura que cada uno de los elementos que están comprendidos en la investigación, se correlaciona entre sí, en otras palabras, que exista congruencia horizontal y vertical entre los componentes medulares de la investigación cualitativa. Esto nos permite tener una secuencia lógica, de manera que cada elemento de la investigación, problema, objetivos, categorías de análisis, y recomendaciones mantengan correlación y realmente den un aporte científico, pertinente y viable. A continuación, se incluye la matriz de congruencia construida para esta investigación:

Tabla 1.1. Tabla Matriz de congruencia

PROBLEMA DE INVESTIGACIÓN	HIPÓTESIS
---------------------------	-----------

<p>En Morelos existen rutas de recolección de residuos sólidos en el sector turístico y de servicios. Estas rutas se determinan de forma empírica y se desea saber si se puede optimizar estas rutas</p>	<p>H1. A través del uso de WCVRP con ventanas de tiempo se puede resolver el problema de recolección de basura en el sector turístico en Cuernavaca; Morelos</p> <p>H0. A través del uso de WCVRP con ventanas de tiempo no se puede resolver el problema de recolección de basura en el sector turístico en Cuernavaca, Morelos.</p>
OBJETIVO GENERAL	
<p>Encontrar una ruta factible para el problema de recolección de residuos sólidos en el sector turístico en Cuernavaca, Morelos.</p>	
OBJETIVOS ESPECÍFICOS	ANÁLISIS RPF
<ul style="list-style-type: none"> ● Comprender el funcionamiento del WCVRP con ventanas de tiempo ● Modelar matemáticamente el WCVRP con ventanas de tiempo ● Revisar la literatura en el estado del arte del WCVRP con ventanas de tiempo en el sector turístico ● Buscar instancias aplicables para la solución del WCVRPTW ● Identificar los algoritmos que tengan el mejor resultado para el problema en cuestión ● Proponer nuevos enfoques heurísticos para resolver este problema. ● Implementar cada uno de los enfoques propuestos. ● Realizar experimentos computacionales para calibración de los parámetros utilizados en las estrategias implementadas. ● Realizar comparación de los resultados al utilizar la heurística propuesta y los resultados obtenidos por heurísticas reportadas en la literatura. ● Simular el problema WCVRPTW de la recolección de residuos sólidos en el sector turístico de Cuernavaca, Morelos ● Verificar con un caso de estudio si la solución propuesta es funcional ● Documentar los análisis, resultados y conclusiones, en un artículo publicable en una revista internacional. 	<p>Relevancia. El problema de la recolección de residuos sólidos es un tema relevante debido a la alta generación de basura y carencia al tratar estos residuos mediante la ayuda de técnicas de optimización.</p> <p>Pertinencia. El WCVRPTW ha sido abordado con métodos de optimización y cómputo aplicado, además de que es un tema de discusión actual en el área de optimización</p> <p>Factibilidad. Se cuenta con instancias de prueba e información real para llevar a cabo todas las simulaciones requeridas</p>

En la **Tabla 1.1**, se describe la matriz de congruencia de la asignación de rutas de recolección de residuos sólidos que sistematiza a la investigación en problemática, objetivos, variables y ejecución práctica de las variables, con ello se sintetiza la idea principal de la investigación de una forma simple, con enunciados concisos acerca del tema principal de la investigación y se identifican las variables reales o los aspectos teóricos bajo investigación y la relación entre ellos. Como conclusión, es decir que la

matriz de congruencia es un cuadro que permite reducir tiempo y esfuerzos destinados a la investigación, pues facilita organizar cada una de las etapas del ciclo y es útil para comprobar la coherencia entre ellas.

1.10. Organización de la tesis

La se encuentra organizada de la siguiente manera: después de investigar el problema de recolección de basura de empresas del sector turístico en el Capítulo 1, este problema pertenece a la clase de enrutamiento vehicular, específicamente al problema de enrutamiento de vehículos (VRP) y sus extensiones. En el Capítulo 2, se presenta el desarrollo del problema de enrutamiento de vehículos y los resultados de investigaciones anteriores como la revisión de la literatura y en el Capítulo 3, Con una comprensión profunda de WCVRPTW, discutimos el problema de recolección de basura de empresas del sector turístico, se adopta el algoritmo de recocido simulado para generar una solución para este problema. Dado que algunos de los clientes tienen ventanas de tiempo para recoger la basura, el problema se extiende al Problema de Enrutamiento de Vehículos Capacitados con Windows de Tiempo (WCVRPTW) agregando restricciones de ventanas de tiempo. En el Capítulo 4 se aplica el algoritmo de recocido simulado para resolver el VRPTW, se realiza un diseño de experimentos para probar la validez del algoritmo que posteriormente mejorará la eficiencia de la distribución de las rutas de recolección del caso de estudio real. Por último, en el Capítulo 5 se extraen conclusiones y se da una recomendación de futuras investigaciones sobre este problema. Finalmente, se presentan las referencias utilizadas.

Capítulo 2. Marco Teórico del problema de enrutamiento de vehículos

En este capítulo se muestran los conceptos y definiciones acerca del problema de enrutamiento de vehículos para la recolección de residuos sólidos con Ventanas de Tiempo. Una descripción del problema de enrutamiento de vehículos y algunas de sus generalizaciones para la contextualización del problema a tratar de este trabajo se muestra con los diferentes tipos de problema de transporte. Se presentan las características del problema abordado, se describe el marco teórico donde se describen los trabajos relacionados con el propósito de facilitar la comprensión de los diversos aspectos involucrados. Posteriormente, se discuten los principales trabajos de la literatura relacionada que abordan el problema. Se presenta la definición formal del WCVRPTW y su formulación matemática. En la segunda parte de este capítulo, se abordan ejemplos de técnicas heurísticas que se han utilizado para resolver el WCVRPTW para la recolección de residuos sólidos.

2.1. Smart Cities

Una ciudad inteligente según (Mohanty, Choppal, & Kougianos, 2016) es un lugar donde las redes tradicionales y los servicios se hacen más flexibles, eficientes y sostenibles con el uso de la información digital y tecnologías de telecomunicaciones para mejorar sus operaciones en beneficio de sus habitantes. Dentro de las ciudades inteligentes, las tecnologías digitales se traducen en mejores servicios públicos para los habitantes y para un mejor aprovechamiento de los recursos sin impactar en el medio ambiente. Una definición formal de una Smart City sería (Mohanty, Choppal, & Kougianos, 2016):

“Una ciudad inteligente sostenible es una ciudad que utiliza técnicas innovadoras mediante las tecnologías de información y la comunicación (TIC) con el fin de mejorar calidad de vida de los habitantes, sin poner en riesgo los recursos actuales para un bienestar de las futuras generaciones”.

Las ciudades inteligentes usualmente utilizan sensores de baja potencia, redes inalámbricas y aplicaciones móviles para medir y optimizar todo dentro de las ciudades. La **figura 2.1** ilustra una descripción general de varios componentes en una ciudad inteligente. Se puede apreciar que dentro de las ciudades inteligentes existe una subdivisión de transporte inteligente para la asignación de rutas rápidas inteligentes y adaptativas.



Figura 2.1. Esquema de Smart Cities (Tech Data Corporation, 2020)

2.2. Transportación Inteligente

Sistemas de transporte tradicionales tales como la red ferroviaria, transporte por carretera, transporte aéreo, y el transporte de agua existe desde hace mucho tiempo. En el transporte tradicional cada uno de estos opera independientemente incluso en un tipo específico del sistema de transporte, lo que dificulta su uso global.

El sistema de transporte ha hecho posible la construcción de centros de vías aéreas globales, redes ferroviarias interurbanas, redes de carreteras inteligentes, rutas ciclistas protegidas, rutas peatonales protegidas y transporte público integrado para un transporte seguro, rápido, rentable y confiable. En este mismo sentido, el uso de las TIC y el procesamiento de datos en tiempo real ha hecho posible que el sistema de transporte sea inteligente. Éste permite a las personas seleccionar fácilmente diferentes opciones de transporte para rutas de bajo costo, distancias más cortas o rápidas, en la **figura 2.2** se ilustra un sistema de transporte inteligente. El sistema de transporte inteligente maximiza la utilización de los vehículos utilizados en el sistema, por ejemplo, el número de aeronaves que una aerolínea tiene, el número de trenes que tiene una red ferroviaria o el número de vehículos utilizados para la recolección de los residuos. Otro ejemplo de transporte inteligente podría ser el uso de aplicaciones inteligentes en teléfonos móviles para asignar los clientes a los vehículos de recolección e incluso rastrear la ubicación exacta de la información del vehículo y del cliente en la misma aplicación inteligente.



Figura 2.2. Problemas de Transporte (Fuente propia).

Se presenta una amplia ilustración de diferentes tipos de sistemas de transporte inteligente como el transporte vehicular, ferroviario, marítimo y aéreo.

2.3. Recolección de residuos

En general, la planificación de la asignación de recolección de residuos implica la recolección y el transporte de residuos sólidos a los lugares de procesamiento. Este servicio esencial está recibiendo cada vez más atención de muchos investigadores debido al impacto ambiental y al crecimiento del número de desechos sólidos debido al aumento de la población especialmente en las áreas urbanas, y el alto costo de su recolección por las que la problemática se ha convertido en un área de investigación importante en el enrutamiento de vehículos. Este servicio implica un costo operacional muy alto, los investigadores están tratando de reducir el costo al mejorar el enrutamiento de los vehículos de recolección de residuos al encontrar la ubicación más adecuada de las instalaciones de eliminación, la ubicación de los contenedores de recolección de residuos, así como minimizar el número de vehículos utilizados.

Esta tesis considera el problema de enrutamiento de vehículos que surge cuando un conjunto de clientes tiene desechos que deben ser recolectados por los vehículos. En tales situaciones, es común que la cantidad de desechos sea tal que los vehículos se llenan durante su jornada laboral y tengan tiempo para visitar una instalación de eliminación, para vaciarse antes de visitar a más clientes y recolectar más desechos. Como tal, se pueden realizar múltiples visitas a las instalaciones de eliminación de residuos durante la jornada laboral. Este problema es un enrutamiento de nodo de un solo período.

En el problema de la recolección de desechos, las organizaciones deben retirar los desechos del punto de recolección y enviarlos a las instalaciones de tratamiento. En este problema, normalmente se deben recopilar muchos puntos de recolección para poder enviarlos a la instalación apropiada (generalmente solo un depósito).

El problema de la recolección de desechos comerciales, formalmente se puede describir de la siguiente manera: el problema tiene un número ilimitado de vehículos idénticos (homogéneos) basados en un solo depósito. Los vehículos comienzan y terminan sus

rutas en el depósito. Este problema involucra múltiples instalaciones de eliminación, por lo que las decisiones deben tomarse no solo sobre cuándo un vehículo debe vaciarse en una instalación de eliminación, sino también qué instalación de eliminación debe usar. Este problema también considera las ventanas de tiempo, una asociada con cada cliente que rige cuándo se pueden recolectar los desechos de ese cliente; otro asociado con cada instalación de disposición que rige cuando esa instalación está abierta; otro asociado con el depósito que rige cuando está abierto para despachar / recibir vehículos. Cada vehículo tiene un período de descanso para el conductor (asociado a una pausa para el almuerzo durante el día laboral) y una cantidad máxima de trabajo que puede realizar durante el día (en términos de la cantidad total de residuos recolectados como del número total de clientes atendidos).

Esta tesis está enfocada a la recolección de residuos sólidos en el sector turístico comercial en Cuernavaca, Morelos, México. Los residuos comerciales se pueden encontrar en restaurantes, puntos de venta y en centros comerciales. Los vehículos recolectarán los desechos en los puntos de recolección antes de ir a las instalaciones de disposición para ser vaciados.

Uno de los principales desafíos del gobierno de México es generar un desarrollo que fomente la calidad de vida para los ciudadanos mexicanos. Uno de esos desafíos se encuentra dentro del área de movilidad del país, para poder tener un desarrollo en dicha área es necesario promover nuevas tecnologías para alcanzar altos estándares. Sin embargo, la realidad de la movilidad de las ciudades del país ha demostrado una deficiencia en sus soluciones, con ello un incremento del tránsito vehicular, tráfico vehicular y consecuencias ambientales. Esta movilidad deficiente y la accesibilidad urbana entorpecidas son factores que afectan la economía del país, por lo que se requiere que las ciudades tomen medidas de forma urgente y firme enfocados a mejorar la vialidad. Es por ello, que México debe fomentar criterios para la mejorar de la vialidad que permita reducir este rezago, con ello disminuir el impacto ambiental y mejorar la movilidad de los habitantes. En la **figura 2.3** se muestra un panorama general de la situación de logística del transporte en México.

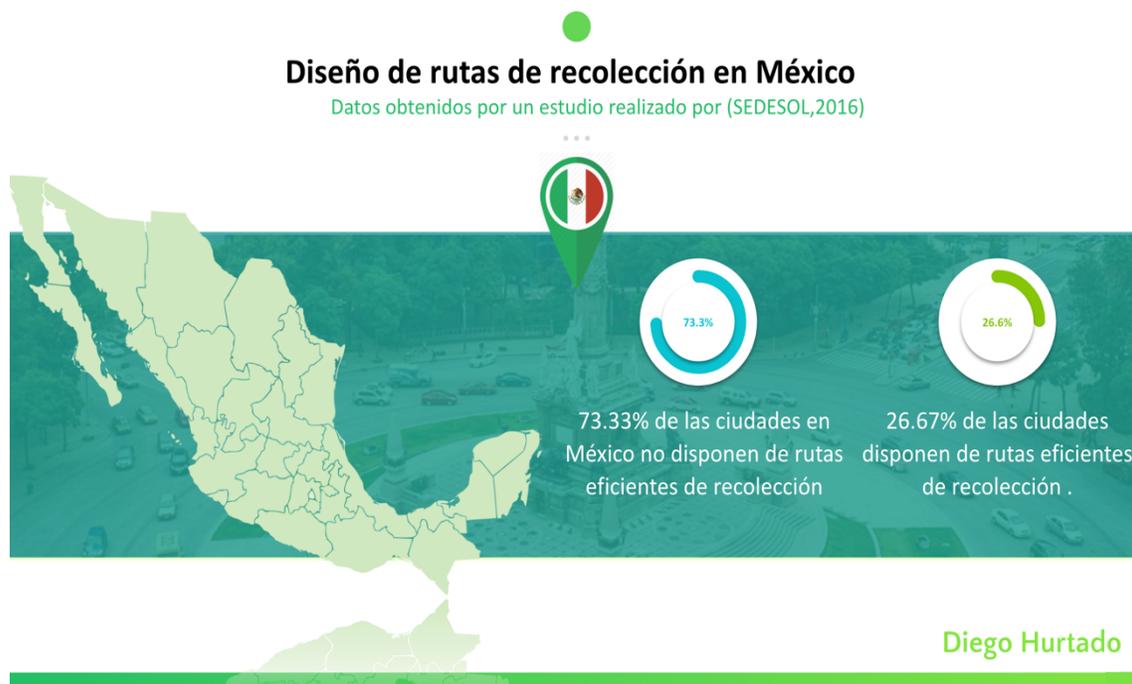


Figura 2.3. Diseño de rutas de recolección en México, fuente (INAPAM, 2017)

En la **Figura 2.3** se muestra un panorama general de la situación de logística de transporte en México. Donde se puede observar que la mayor parte de las rutas de recolección de residuos en México no disponen de rutas eficientes, a pesar de que el manejo de los desechos es un tema relevante dentro de las ciudades por el impacto en la salud pública de los habitantes, sin embargo, no se están tomando las medidas necesarias para mejorar las rutas de recolección.

En la siguiente sección, se revisan la clasificación de residuos sólidos. Primeramente, tomamos como base el enfoque propuesto por (Kim et al., 2006) al problema de enrutamiento de Vehículos para recolección de basura con ventanas de tiempo (WCVRPTW) y lo adaptamos a nuestra problemática. Debido a que (Kim et al., 2006) han hecho públicas sus instancias de prueba reales, su trabajo ha sido reconocido. Comparamos nuestro enfoque con el de (Kim et al., 2006) debido a que para futuras investigaciones podamos tomar en cuenta las mismas variables para instancias muy grandes como las que él está tomando en cuenta. Una diferencia importante entre el problema de recolección de Kim con el nuestro es que este cuenta con instalaciones de

eliminación y el problema de entrega con las instalaciones de reabastecimiento, es que en el problema de recolección un vehículo visita una instalación de eliminación para vaciarse inmediatamente antes de regresar al depósito.

2.3.1. Clasificación de residuos

Según un estudio realizado por (INAPAM, 2017), la recolección de los desechos se puede clasificar en: fuentes domiciliarias, sectores comerciales, sitios públicos, instituciones, sector hospitalario y sector industrial. Sin embargo, existe otra clasificación para el manejo de residuos: residuos municipales y residuos especiales. Primeramente los residuos municipales son aquellos que se generan en casa habitación, mercados, comercios, instituciones, parques y vías públicas, es decir todos aquellos que no necesitan de un tratado en especial, por otra parte los residuos especiales son aquellos que necesitan un manejo especial como los residuos generados por procesos industriales hospitales, laboratorio, actividades agrícolas y actividades nucleares, es decir por sus características físicas, químicas y biológicas deben ser manejados, tratados y dispuestos utilizando métodos adecuados para evitar riesgos a la salud y a la ecología. Usualmente la mayor parte de los estudios y la información sobre su manejo son realizados en cuestión de la gestión de residuos domiciliarios generados en casas habitación y comerciales, existiendo muy poca información acerca de los residuos generados en otras fuentes como las industriales. En este sentido, los siguientes datos obtenidos por un estudio del (INAPAM, 2017) nos dan un panorama general de la situación actual de la recolección, dentro de las ciudades de México: “El 60.97% de los residuos sólidos generados proceden de fuentes domiciliarias y el restante 39.02% de las industrias, comercios y otras fuentes”.

2.3.2. Aspectos relacionados con los costos monetarios

La etapa de recolección de residuos emplea un número considerable de recursos económicos en las ciudades (INAPAM, 2017). Uno de los principales factores que se consideran para las decisiones de distribución de la recolección de residuos sólidos es el

costo de transporte. Este costo se puede calcular en km o en tiempo (horas), que corresponden al costo total de los vehículos que viajan desde el origen (depósito) a los destinos de recolección (clientes), luego regresando al depósito de partida. En problemas de enrutamiento de vehículos, estos costos se consideran en términos de distancia (o tiempo). Tenga en cuenta que estos valores también se pueden utilizar para calcular otros indicadores, como la contaminación o en algunos casos. Un dato importante es que la congestión del tráfico y la contaminación atmosférica asociada se consideran la amenaza más importante para países como el Reino Unido (, especialmente para la industria del turismo, ya que dejan una impresión negativa en los visitantes Schadschneider, Klüpfel, Kretz, Rogsch, & Seyfried, 2009). Actualmente, Sin embargo, este no es el único costo que tiene que ser considerado en problemas de transporte, a éste tendría que agregarse el costo de las soluciones de infraestructura y tecnológicas. Estas nuevas alternativas de innovación hacen referencia a sistemas que conllevan un esfuerzo y retos específicos, en el cual los gobiernos deben de asumir el cargo para poder hacer realidad estas mejoras a través de regulaciones, políticas, leyes o medidas fiscales con las cuales se puedan atacar dichos rezagos directamente. Bajo este principio, es necesario realizar mejoras en la vialidad para que los vehículos puedan movilizarse de un lugar a otro de una forma más eficiente (Castán, Ibarra, Laria, Guzmán & Castán ,2014). La construcción del sistema de transporte también puede ser importante, dependiendo de los fondos que las autoridades disponen para un sistema de logística urbana. Otros costes que también podrían ser considerados son los costos de ejecución y los costos de mantenimiento, que deben ser asumidos por el sistema de transporte en la óptica a largo plazo para asegurar que el sistema sea autosuficiente y pueda ser operativo sin la ayuda de fondos públicos. Hoy en día las ciudades del primer mundo enfrentan una infinidad de retos, uno de ellos es el control de tráfico vehicular que es uno de los más complejos y significativos. De ahí, que una de las principales inquietudes de los administradores y representantes de las ciudades modernas es lograr el control óptimo de sus vialidades. Por consiguiente, este tipo de mejoras para tener un mejor manejo de control vehicular promete nuevas oportunidades de mejora en la calidad de vida de sus habitantes, sin embargo, esto conllevaba muchos retos tecnológicos para poder hacerlo

realidad. Por último, otro costo sería el salario de las horas trabajadas por las personas que se encargan de realizar la recolección.

2.3.3. Aspectos ambientales: contaminación del aire y suelo

Uno de los principales objetivos de las ciudades es disminuir la contaminación del aire. En las últimas décadas, la cantidad de automóviles ha aumentado. El 95% del consumo de gasolina en México es destinado al autotransporte (Solís & Sheinbaum, 2016). Los cambios en los hábitos de vida de los habitantes han aumentado el uso de los automóviles privados, con ello la congestión del tráfico y la contaminación del aire. Podemos observar diferentes sustancias contaminantes, que están directamente relacionadas con el combustible. A nivel urbano los vehículos son la principal fuente de contaminantes. En México 95% de las emisiones de CO_2 , el 73% del NO_x y el 15% de SO_2 se estima que proviene de las emisiones producidas por los vehículos (INE, 2010a). Primeramente, el CO_2 , es una de las principales causas del calentamiento global. En México los vehículos particulares generan en promedio el 18% de las emisiones de CO_2 , un dato importante es que esto causa un costo muy grande, si no se toman las medidas de prevención adecuadas al país le podría costar hasta el 6% del PIB (ONU-Hábitat, 2015).

Un dato relevante sobre la calidad del aire en México es que “más de 34 millones de personas están expuestas a mala calidad del aire”, la mayor parte de esta contaminación es debido en a los automóviles (INAPAM, 2017).

Datos de estudios realizados estiman que “14,734 muertes están relacionadas con la mala calidad del aire en 2008” según la Organización Mundial de la Salud. La mejor solución para reducir la contaminación parece ser una combinación de organización y estrategias tecnológicas (Cabrero, Orihuela, & Ziccardi, 2003). Muchos países optan por realizar estas tecnologías, lo cual muestra una reducción considerable tanto de la congestión como de la contaminación. Es por ello por lo que se debe tomar cartas en el asunto para implementar nuevas normas y políticas, con el fin de disminuir la cantidad de kilómetros recorridos por los automóviles mejorando así el flujo vehicular; esto se

considera una solución posible, real, viable y que se debería implementar en nuestro país como una nueva alternativa moderna para mejorar la calidad del transporte en México. Por lo anterior, es necesario que México cree ciudades sustentables, que implementen nuevas técnicas para la mejora de la calidad de vida de los habitantes (Cabrerero, Orihuela, & Ziccardi, 2003). Sería importante que México pudiera implementar estos tipos de tecnologías para mejorar la transportación y con ello reducir la contaminación y el tráfico vehicular. En la **figura 2.4** se muestra la situación actual de la ciudad de México, en ella se puede apreciar que los edificios que perfilan la ciudad se ocultan en una nube oscura que apenas deja ver el sol.



Figura 2.4. Contaminación en ciudad de México (López Doriga, 2015)

2.4. Enrutamiento de vehículos enfocado a la recolección de residuos sólidos

Tratar un problema de enrutamiento de vehículos es diferente del problema de enrutamiento de vehículos para recolección como se discutió en la sección anterior porque hay una restricción adicional que debe ser considerada para resolver este problema. En lugar de regresar al depósito para descargar los productos recolectados, en un problema de recolección de desechos, los vehículos deben ser vaciados en un centro

de disposición antes de continuar recolectando desechos de otros clientes. Por lo tanto, se producen varios viajes al centro de disposición antes de que los vehículos regresen a los depósitos vacíos, con cero desperdicios. Una complicación en el problema surge cuando se trata de más de un centro de recolección. Aquí, se necesita determinar el momento adecuado para vaciar los vehículos, así como elegir el mejor centro de distribución a la que deben ir para que la distancia total se pueda minimizar. Otra consideración que puede que no sea óptimo permitir es que el vehículo de recolección se llene antes de visitar una instalación de eliminación. Por lo que es de suma importancia conocer la localización de los vehículos, en ese sentido los sistemas computarizados de navegación para automóviles son probablemente la aplicación más popular de los sistemas de información geográfica, por una red que suele estar representada por un grafo, como se muestra en la **figura 2.5**.

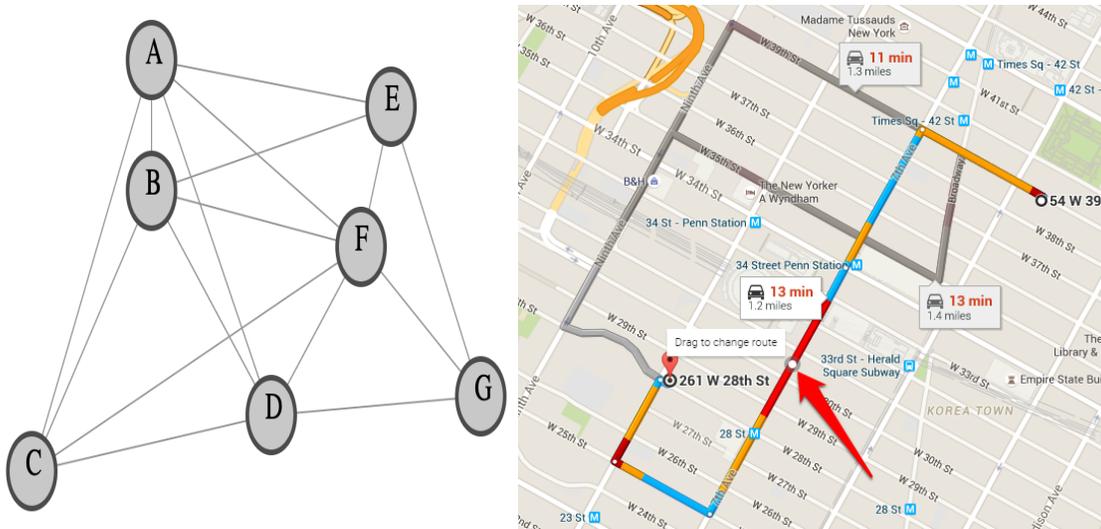


Figura 2.5. Representación de red de clientes, usualmente suele estar presentada por un grafo (Google Developers, 2020)

Una representación esquemática de la red de calles es generalmente proporcionada por grafos como se muestra en la figura de arriba en el lado izquierdo. Los sistemas de navegación para automóviles se hicieron posibles con la construcción de grafos de las redes de calles (figura de arriba, lado derecho). Para los propósitos actuales, un grafo es una estructura, que consta de nodos y aristas, relacionados por incidencia y adyacencia.

Cada nodo representa una ubicación y una ruta es un camino a través de un conjunto de nodos. La mayoría de los problemas de enrutamiento reales implican encontrar rutas eficientes para vehículos (automóviles, trenes, aviones, etc.), por lo que a menudo se los denomina problemas de enrutamiento de vehículos. Este se puede describir como el problema de diseñar una entrega óptima o las rutas de recolección de uno o varios depósitos a una serie de ciudades o clientes geográficamente dispersos, sujeto a restricciones laterales. Los problemas de enrutamiento se pueden dividir en dos tipos principales: problemas de enrutamiento de nodos y problemas de enrutamiento de arco, dependiendo de si el objetivo es visitar los nodos (ubicaciones) o los arcos (los bordes que los conectan) (Fukasaw, Longo, Lysgaard, De Aragão, Reis, Uchoa, & Werneck, 2006).

En la **figura 2.6** se muestra la evolución del problema de transporte desde sus primeros inicios con el problema de agente viajero hasta el WCVRPTW.

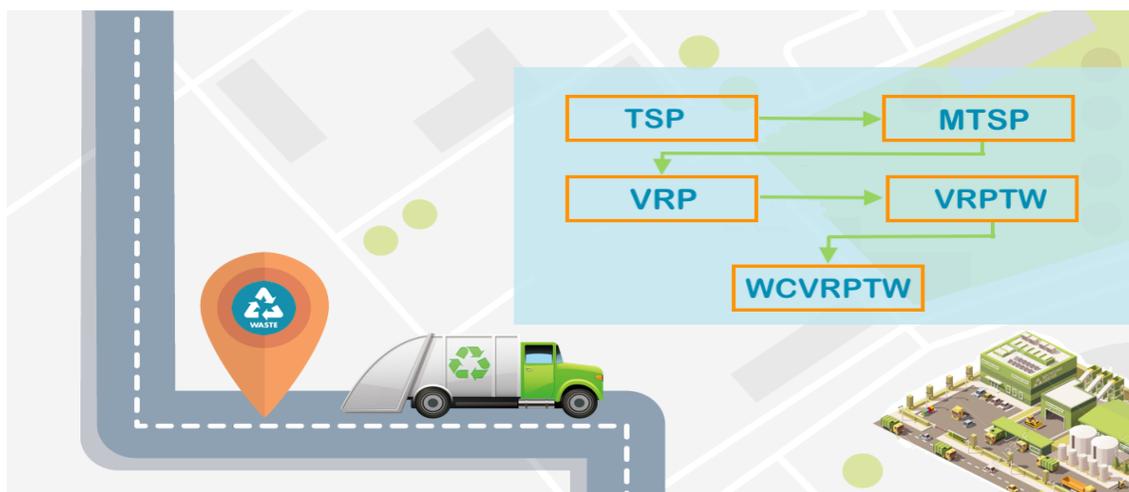


Figura 2.6. Evolución del problema de WCVRPTW

Cada una de las variantes incluye más restricciones, la primera variante fue el TSP, y dependiendo de la problemática a resolver se hace referencia a la variación del problema.

2.5. Teoría de grafos

La formulación de problemas de enrutamiento de vehículos (VRP) están basados en grafos, y los algoritmos de optimización se desarrollaron como una técnica para resolver problemas computacionales que pretenden encontrar buenos caminos a través del grafo. Por lo tanto, el conocimiento de la teoría de grafos ayudará a comprender tanto el problema como su solución en un nivel más preciso. A continuación, definiciones básicas de la teoría de grafos se presentarán, algunas de ellas se toman del libro de (Horowitz, Sahni & Rajasekaran, 1997).

Un grafo (finito) $G = (V, E)$ consiste en dos conjuntos finitos V y E . Se llama un elemento v de V un vértice y un elemento e de E , a un par de vértices, se les llama aristas. Si uno quiere enfatizar que un conjunto de vértices V pertenece al grafo G , se puede escribir $V(G)$; de manera similar, si uno quiere enfatizar que un conjunto de las aristas E pertenece a G , se puede escribir $E(G)$. Si cada arista en $E(G)$ es un par desordenado de dos vértices, el grafo G se dice que no está dirigido; si cada borde de $E(G)$ es un par ordenado de dos vértices, se dice que el grafo G es un grafo dirigido. El orden de un grafo es la cardinalidad de su conjunto de vértices; El tamaño de un grafo es la cardinalidad de su conjunto de aristas.

Sea que $e \in E(G)$ y $e = \{u, v\}$ para algunos $u, v \in V(G)$. Por conveniencia de notación, generalmente se denota simplemente como u_v . Los vértices u y v se llaman vértices finales de e , y se dice que son adyacentes. Además, se dice que u (similarmente para v) es incidente con e . Si $\{u, v\} \notin E(G)$, entonces los vértices u y v no son adyacentes. Si $u = v$, entonces u_v se llama un bucle. La vecindad de un vértice v , denotado por $N(v)$, es el conjunto de vértices adyacentes a v . La vecindad de un conjunto S , denotado por $N(S)$, es la unión de las vecindades de los vértices en S . El grado de un vértice v , denotado por $\text{grado}(v)$, es el número de aristas incidentes con v . Se dice que un grafo está completo si cada vértice es adyacente a cada otro vértice.

Un recorrido en un grafo, también conocido como tour es una secuencia de vértices V_1, V_2, \dots, V_n tal que $V_i + 1 \in E$ para todo $i = 1, 2, \dots, n - 1$, y V_1 y V_n se denominan los vértices finales del camino. Si todos los vértices de un recorrido o tour son distintos, el recorrido se llama camino. Si todas las aristas en un camino son distintas, el recorrido se llama camino. Un camino cerrado, o un ciclo, es un camino v_1, v_2, \dots, v_n ($k \geq 3$) junto con un borde V_k . Un sendero que comienza y termina en el mismo vértice se llama un camino cerrado, o un circuito. La duración de un recorrido (o un camino, un camino, un ciclo, un circuito) es el número de bordes, contando repeticiones. Un grafo conectado es denominado si cada par de vértices puede unirse por un camino.

Un camino hamiltoniano (**figura 2.7**) es un camino que contiene cada vértice de G . De forma similar, un ciclo hamiltoniano es un ciclo que contiene cada vértice de G . El grafo g se llama hamiltoniano si contiene al menos un ciclo hamiltoniano (Horowitz, Sahni, & Rajasekaran, 1997).

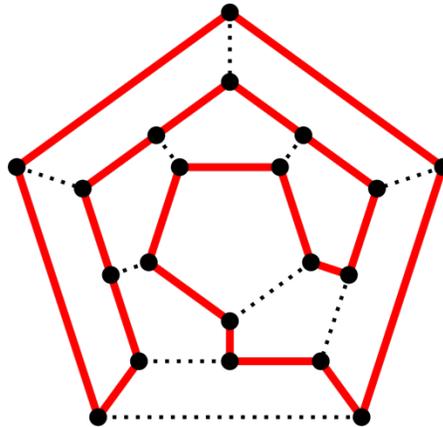


Figura 2.7. Ciclo Hamiltoniano

Un ciclo Hamiltoniano en un grafo dirigido $G = (V, E)$ es un ciclo dirigido de longitud $n = |V|$. Entonces, el ciclo recorre cada vértice exactamente una vez y luego regresa al vértice inicial.

2.5.1. Matriz de adyacencia

Aunque existen varias representaciones para resolver el problema de enrutamiento de vehículos, estudiamos la más comúnmente utilizada, la matriz de adyacencia que se muestra en la **figura 2.8**.

Sea $G = (V, E)$ un grafo con n vértices, $n > 1$. La Matriz de Adyacencia de G es un arreglo bidimensional $n \times n$, es decir a , con la propiedad que $a[i, j] = 1$ si la arista (i, j) ((i, j) para un grafo dirigido) está en $E(G)$. El espacio necesario para representar un grafo utilizando su matriz de adyacencia es n^2 bits.

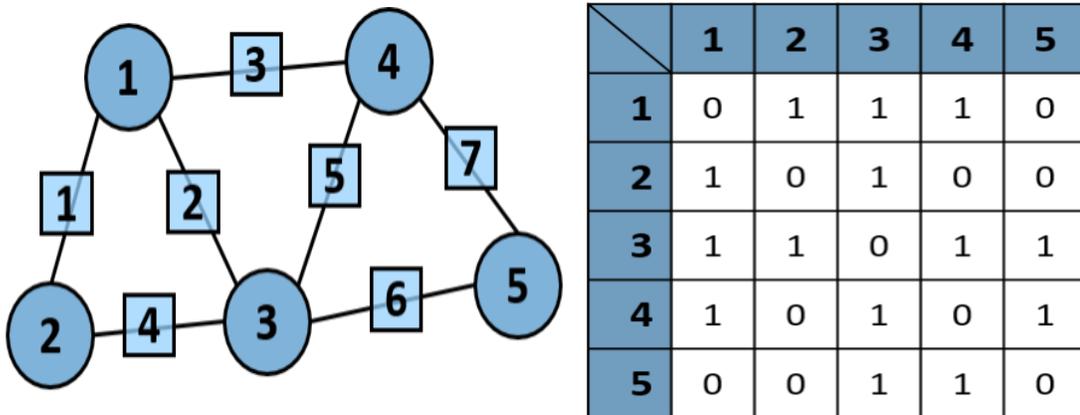


Figura 2.8. Matriz de adyacencia

Desde la matriz de adyacencia, podemos determinar fácilmente si hay un borde que conecta dos vértices i y j . Para un grafo no dirigido, el grado de cualquier vértice es su suma de filas se representa en la ecuación (2.1):

$$\sum_{j=1}^n a[i, j] \tag{2.1}$$

Para un grafo dirigido, la suma de la fila es el grado de salida, y la suma de las columnas es el grado de entrada. Para saber el grado o el número de aristas de un grafo o si un grafo está conectado, las matrices de adyacencia requieren al menos n^2 tiempo, $n^2 - n$ debido a las entradas de la matriz (las entradas diagonales son cero).

2.6. Optimización Combinatoria

De acuerdo a (Grotschel, & Lovász, 1995) optimizar significa encontrar el máximo o mínimo de una determinada función, definido en algún dominio. Las teorías clásicas de optimización (cálculo diferencial, cálculo de variación, teoría de control óptimo) tratan el caso cuando este dominio es infinito. Desde este ángulo, el tema de la optimización combinatoria, donde el dominio es típicamente finito, puede parecer trivial: es fácil decir que "elegimos lo mejor de este número finito de posibilidades". Pero, por supuesto, estas posibilidades pueden incluir todos los árboles en un nodo. Los problemas de optimización combinatoria generalmente tienen como entradas números y estructuras combinatorias (por ejemplo, un grafo con pesos en los bordes). En el modelo de máquina de Turing, ambos están codificados como cadenas 0-1. A menudo, cuando la función objetivo es demasiado compleja, las restricciones son demasiado complicadas o el tamaño del problema es demasiado grande, es imposible encontrar una solución óptima. Con frecuencia, esto no es solo una experiencia práctica; las matemáticas y la informática han desarrollado teorías para hacer afirmaciones intuitivas sobre la dificultad de ciertos problemas precisos.

2.7. Complejidad

Según (Lageweg, Lawler, Lenstra, & Rinnooy, 1981), el VRP y el VRPTW pertenecen al conjunto de problemas NP de optimización combinatoria NP-duro. El VRPTW en particular, según (Landeghem, 1988) tomando en cuenta las restricciones de ventanas de tiempo sigue siendo mucho más complicado de resolver que un simple VRP incluso para una flota fija de vehículos. Por eso cuando se requiere resolver este tipo de problemas se sugieren principalmente procedimientos heurísticos para resolver instancias de problemas más grandes del VRPTW.

Los métodos heurísticos a menudo producen soluciones óptimas o casi óptimas para problemas grandes se puede resolver en un periodo de razonable de tiempo de

cómputo. Por lo tanto, el desarrollo de algoritmos heurísticos que pueden obtener soluciones factibles casi óptimas es una forma viable de resolver el problema. Sin embargo, con instancias grandes con 100 clientes o más son muy difíciles de resolver de manera óptima. Sin embargo, muy pocos de los puntos de referencia de las instancias de Solomon que involucran a 100 clientes se han resuelto de manera óptima (Fisher, Jörnsten, & Madsen, 1997) y (Kohl, 1995).

La fórmula para calcular el número de posibles combinaciones se muestra en la ecuación (2.2), donde n representa el número de clientes y -1 representa la disyunción del nodo depósito.

$$(n-1)! = 1 \times 2 \times 3 \times \dots \times n \tag{2.2}$$

En la **figura 2.9** se muestra una red simétrica: la red asimétrica tiene una ruta única para el tráfico de red entrante y saliente. Como tal, el tráfico toma la misma ruta al entrar o en la red. Es decir que es el mismo costo de ir de A - B que de B - A.

Cientes	Número de rutas
3	3
4	12
5	60
6	360
7	2520
8	20160
9	181,440
10	1,814,400

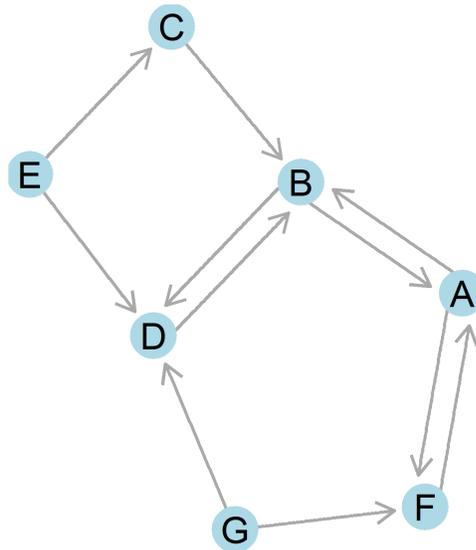


Figura 2.9. Complejidad del VRPTW Simétrico (Lenstra & Kan, 1981)

En la **figura 2.10** se muestra una red asimétrica: una red asimétrica tiene múltiples rutas para el tráfico de red entrante y saliente. Como tal, el tráfico toma una ruta diferente al entrar o salir de la red.

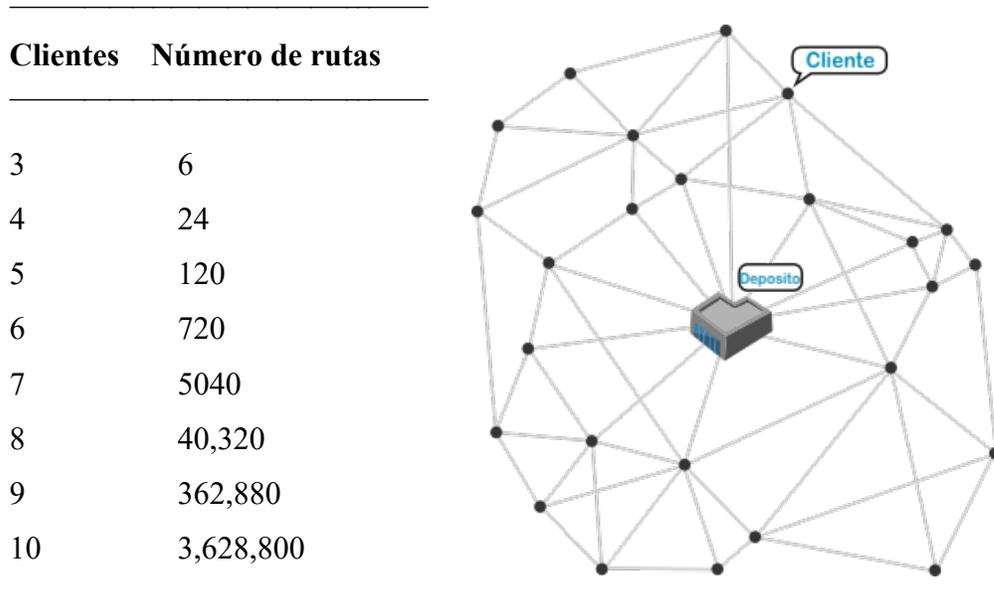


Figura 2.10. Complejidad del VRPTW no Simétrico (Lenstra & Kan, 1981)

2.8. Métodos exactos

Se han propuesto muchas familias de algoritmos exactos para resolver el VRP con una estructura de costos simétrica. Estos se basan en programación lineal entera (PLE), programación dinámica, ramificación y corte (branch-and-bound) Toth & Vigo (1998). Este enfoque propone calcular todas las soluciones posibles hasta que se alcance una de los mejores. Los enfoques exactos más comunes para resolver el VRP han sido utilizados en casos con hasta 135 clientes, pero dentro de un tiempo de computación razonable (es decir, algunas horas en un PC común) generalmente pueden resolver instancias con hasta 100 clientes (Cordeau, Toth & Vigo, 1998). Siguiendo la encuesta de (Laporte & Nobert, 1987), los algoritmos exactos para el VRP se pueden clasificar en tres categorías amplias: (i) métodos de búsqueda directa de árboles; (ii) programación dinámica, y (iii) programación lineal entera. Los algoritmos exactos para resolver VRP,

especialmente el VRP capacitado (CVRP) incluyen los algoritmos de ramificación, ramificación y corte, ramificación y precio. (Gutiérrez-Jarpa, Desaulniers, Laporte, & Marianov, 2010) propusieron un algoritmo exacto para el problema de enrutamiento de vehículos con ventanas de tiempo (VRPTW). Propusieron un algoritmo exacto de ramificación y precio para resolver el problema de enrutamiento de vehículos con ventanas de tiempo.

2.9. Heurísticas clásicas

Los métodos heurísticos realizan una exploración relativamente limitada del espacio de búsqueda y, por lo general, producen soluciones de buena calidad en tiempos de computación modestos. Las heurísticas clásicas para el VRP se dividen naturalmente en Heurísticas constructivas y heurísticas de mejora. El calificativo que hace referencia a "clásico" se refiere al hecho de que la mejora pasos de estas heurísticas realizan descensos, es decir, siempre pasan de una solución a una mejor en su vecino hasta que no haya mejor resultado posible. Los métodos heurísticos para el VRP a menudo se pueden derivar de procedimientos derivados del TSP (Laporte, 1992). Estos procedimientos pueden aplicarse al VRPs y DVRPs casi sin modificaciones. Sin embargo, al aplicar estos métodos para los VRP deben tomarse con cuidado para garantizar que solo se creen rutas factibles para los vehículos. (Chen, Hsueh, & Chang, 2006) proponen un problema de enrutamiento de vehículo tomando en cuenta el tiempo real con ventanas de tiempo. Este problema se formula como una serie de modelos de programación de enteros mixtos que tienen en cuenta los tiempos de viaje en tiempo real y dependientes del tiempo, así como las demandas en tiempo real en un marco unificado. Las rutas de vehículos y los horarios de salida se tratan como variables de decisión, con salida retrasada permitida en cada nodo atendido. (Nagata, & Bräysy, 2009) proponen una construcción de ruta y una heurística de mejora de ruta con heurística de minimización de ruta para el problema de enrutamiento del vehículo con ventanas de tiempo. La heurística propuesta por ellos se basa en un método de inserción y las estrategias de búsqueda local guiada.

2.10. Metaheurística

Se ha visto un progreso significativo en los últimos 15 Años en el desarrollo de metaheurísticas para resolver el VRP (Cordeau et al, 2002). Todos permiten la exploración del espacio de solución más allá del primer mínimo local encontrado. Se han propuesto una gran variedad de esquemas, pero estos pueden clasificarse en tres categorías: (1) búsqueda local, (2) búsqueda de población, y (3) mecanismos de aprendizaje máquina. Muchas de las metaheurísticas más exitosas para las grandes instancias de VRPTW se basan en alguna forma de cálculo paralelo. (Blanton & Wainwright, 1993) fueron los primeros en aplicar un algoritmo genético al VRPTW. Hibridaron un algoritmo genético con una heurística codiciosa. Bajo este esquema, el algoritmo genético busca un buen orden de los clientes, mientras que la heurística codiciosa realiza la construcción de la solución factible. Durante los últimos años, se han escrito numerosos artículos sobre la generación de buenas soluciones para la resolución del VRPTW con el uso de algoritmos genéticos. Casi todas estas investigaciones presentan hibridaciones de un algoritmo genético con diferentes heurísticas de construcción (Berger, Barkaoui & Bräysy, 2003), búsquedas locales (Thangiah, Osman & Sun, 1994) y otras metaheurísticas como la búsqueda tabú (Ho, Ang & Lim, 2001) y el sistema de colonias de hormigas (Berger, Barkaoui & Bräysy, 2003).

2.11. Origen del VRP

Dantzig y Ramser lo propusieron por primera vez en 1959 como el problema de ruteo de camiones, consiste en cómo una flota de camiones homogéneos podría atender la demanda de petróleo de varias estaciones de servicio desde un eje central y tal que la distancia recorrida sea la mínima. El primer registro en la literatura VRP, estudió un TSP a escala relativamente grande y propuso un método de solución. Ese estudio fue seguido por un gran volumen de otros artículos de TSP. El TSP se puede mostrar para ser un caso específico de VRP. (Clarke y Wright, 1964) incorporaron por primera vez más de un vehículo en la formulación del problema.

Primeramente la problemática se modela como “el problema de Ruteo de Vehículos” (VRP) (Dantzig y Ramser ,1959) , por sus siglas en inglés “Vehicle Routing Problem” , un problema comúnmente conocido en logística, transporte y en optimización, consiste en establecer premisas que describen el problema como el número de puntos o nodos, el número de vehículos, distancias, recorridos, depósitos para que finalmente se obtengan rutas más cortas optimizando la distancia y el tiempo de recorrido, es decir, obtener soluciones factibles, este problema tiene aplicación en diversas áreas como redes de computadoras, internet, logística, transporte y redes neuronales por nombrar algunas (Eksioglu et al, 2009).

En cuanto a las aplicaciones del VRP (Rizzoli, Oliverio, Montemanni & Gambardella, 2004): recolección y entrega de mercancías, recolección de basura, limpieza de calles, ruteo del autobús escolar, sistemas dial-a-ride, transporte de personas con discapacidad, ruteo de vendedores. El VRP se ha convertido en un área de investigación cada vez más interesante debido a su amplia aplicabilidad y su importancia económica para reducir los costos operativos en el entorno de los sistemas de distribución. El estudio realizado por (Eksioglu et al, 2009) reveló 1021 artículos con el VRP como el tema principal, publicados entre 1959 y 2008. Según (Eksioglu et al., 2009), la literatura de VRP ha ido creciendo a una tasa del 6% anual.

Los problemas de distribución de bienes a clientes de uno o más depósitos, en general se convierten en problemas sustanciales para muchas empresas o fábricas. Estos problemas de distribución usualmente se han modelado utilizando TSP (El problema del agente viajero) (Flood, 1956) o VRP (problema de enrutamiento de vehículos) y sus variantes. El modelado de las variantes está destinado a tener los modelos más adecuados para específicos problemas de la vida real sobre problemas de distribución. El problema se puede modelar utilizando la teoría de grafos a través del problema del vendedor ambulante (TSP) y Problema de enrutamiento del vehículo (VRP). Algunas restricciones que aparecen en los problemas de distribución son restricciones del cliente, número de depósitos, el número de demandas diversas, la capacidad limitada de los vehículos, que finalmente conducen a la ruta ineficiente. Para ajustar con restricciones,

se pueden construir variantes del TSP. Por ejemplo, una variante del TSP con restricciones adicionales de cada punto (cliente) visitado que tiene una ventana tiempo es TSP con ventanas de tiempo (TSPTW). Otra variante del TSP con restricciones adicionales del punto de inicio y el punto de destino que puede ser más de un depósito es el Múltiple TSP (MTSP) (Tillman, 1969). Estas variantes se desarrollan para modelar aplicaciones en el mundo real que se ajusta a los requisitos específicos. En el siguiente apartado, las variantes de TSP, (mTSP), VRP, VRPTW y WCVRPTW son explicadas a detalle.

El TSP se puede ampliar a problemas de enrutamiento de vehículos (VRP). El VRP puede ser definido como el problema de búsqueda de la ruta óptima para entregar mercancías desde uno o más depósitos a los clientes, o tomar mercancías de un cliente a uno o más depósitos, que cumplan una serie de restricciones de capacidad. El VRP, con una serie de nuevas restricciones, como los costos de viaje, una serie de depósitos, el tiempo de entrega y cualquier otro tipo de restricción adicional. Estas variantes se desarrollan para modelar VRP aplicaciones para ser mejor según los requisitos y limitaciones necesarios. Un número de las variantes del VRP identificadas incluyen:

- Problema de enrutamiento del vehículo capacitado (CVRP) donde cada vehículo tiene capacidad uniforme con un solo producto (Kumar, & Panneerselvam, 2012)
- Múltiple Problema de enrutamiento del vehículo (MDVRP), que es VRP con más de un depósito que sirve a clientes (Chávez, Escobar & Echeverri, 2016).
- El problema de enrutamiento de vehículos con Backhauls (VRPB) es VRP en el que los clientes pueden solicitar una de las entregas o recogidas con el requisito de que en cada vehículo ruta, la recolección se realiza después de que todas las entregas a los clientes se hayan completado (Kumar & Panneerselvam, 2012) .

- Problema de enrutamiento del vehículo con ventanas de tiempo (VRPTW) que agrega ventanas de tiempo en cada cliente para recibir los bienes. (Cordeau,2000)

Para facilitar el uso de varias variantes del VRP que sean adecuadas para los problemas, las modificaciones y la combinación de varios métodos de TSP y variantes de TSP, se presentan a detalle cada una de las variantes en el siguiente apartado.

2.11.1. Problema del agente viajero (TSP)

Los problemas de enrutamiento y planificación a menudo requieren la determinación de secuencias óptimas sujetas a un conjunto dado de restricciones. El problema más conocido de este tipo es el clásico problema del agente viajero (TSP- The Traveling Salesman Problem) introducido por (Flood, 1956). Sea un ciclo Hamiltoniano de costo mínimo en un grafo dado, el cual busca una ruta de longitud total mínima que visita cada uno de los N puntos en un determinado conjunto. Sea $V = \{1, 2, \dots, N\}$ exactamente una vez a través del conjunto del arco A . La distancia entre todos los puntos de combinaciones en A , (i, j) , donde $(i, j) \in V \mid i \neq j$ es conocido. En la **figura 2.11** se muestra un ejemplo de problema del agente viajero en un mapa de estados unidos.

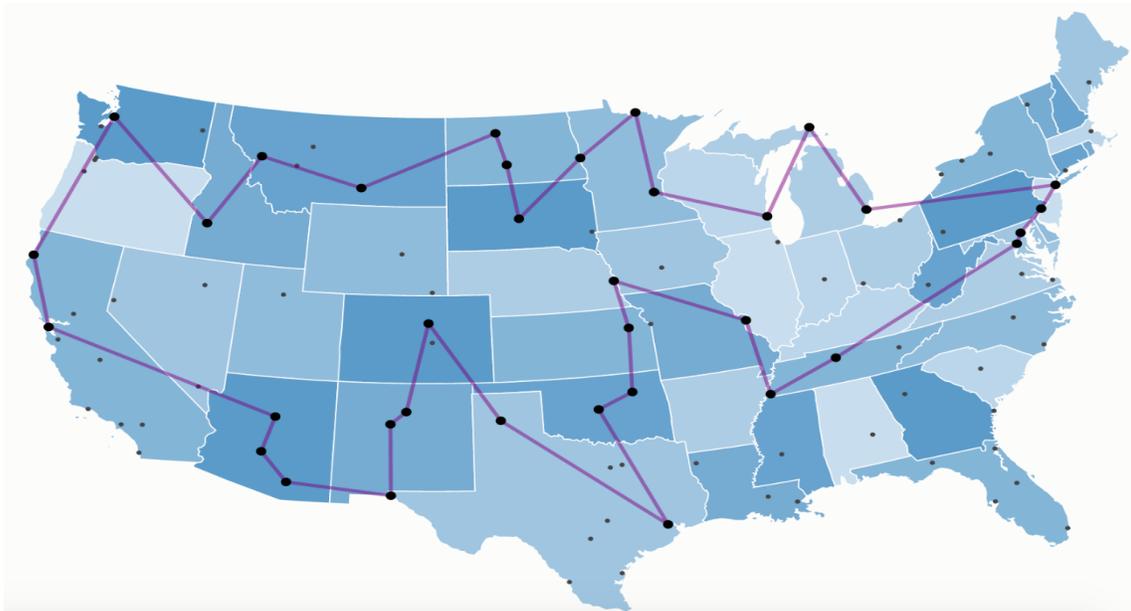


Figura 2.11. Ilustración del Problema de agente viajero (Scilab, 2020)

En la notación introducida para el TSP, la variable X_{ij} definida en la ecuación (2.3) se define de la siguiente manera:

$$X_{ijk} = \begin{cases} 1 & \text{si un vendedor viaja del nodo } i \text{ al nodo } j, \text{ donde } i, j = \{1, 2, \dots, N\} \\ 0 & \text{De lo contrario} \end{cases} \quad (2.3)$$

Definimos formalmente el problema del agente viajero matemáticamente de la siguiente manera (Flood, 1956):

$$\min z = \sum_{(i,j) \in A}^n C_{ij} X_{ijk} \quad (2.4)$$

S.A

$$\sum_{i=1}^n X_{ij} = 1 \quad \forall j \in \{2, \dots, N\} \quad (2.5)$$

$$\sum_{j=1}^n X_{ij} = 1 \quad \forall i \in \{2, \dots, N\} \quad (2.6)$$

$$X_{ij} \in \{0,1\} \quad \forall i \in \{2, \dots, N\} \quad (2.7)$$

La ecuación (2.4) minimiza el costo total de viaje, bajo las siguientes restricciones. El costo total se puede interpretar como el recorrido total en unidades distancia, tiempo o costo. Cada nodo debe ser visitado exactamente una vez de acuerdo con ecuación (2.5) y (2.6), también conocidos como restricciones de grado. La ecuación (2.7) garantiza que el modelo lineal se defina como un modelo lineal de no negatividad tomando un modelo lineal entero de variables binarias.

Aunque existe una serie de variaciones de TSP, nuestro interés está en la variante donde múltiples vendedores son enrutados simultáneamente.

2.11.2. El problema del múltiple agente viajero (MTSP)

El MTSP (The Multiple Traveling Salesman Problem) es similar al TSP notoriamente (**figura 2.12**) con la misma dificultad que busca un recorrido óptimo de N ciudades (Tillman, 1969), visitando cada ciudad exactamente una vez sin sub-tours. En el MTSP, las N ciudades se dividirán en M tours, y cada tour resultará en un TSP para un vendedor. El MTSP es más difícil que el TSP porque requiere determinar qué citas asignar a cada vendedor, así como la ordenación óptima de las ciudades dentro para cada vendedor (Kara, Laporte y Bektas, 2004). Aunque existe un amplio cuerpo de literatura para el TSP y el VRP, el MTSP no ha recibido la misma atención.

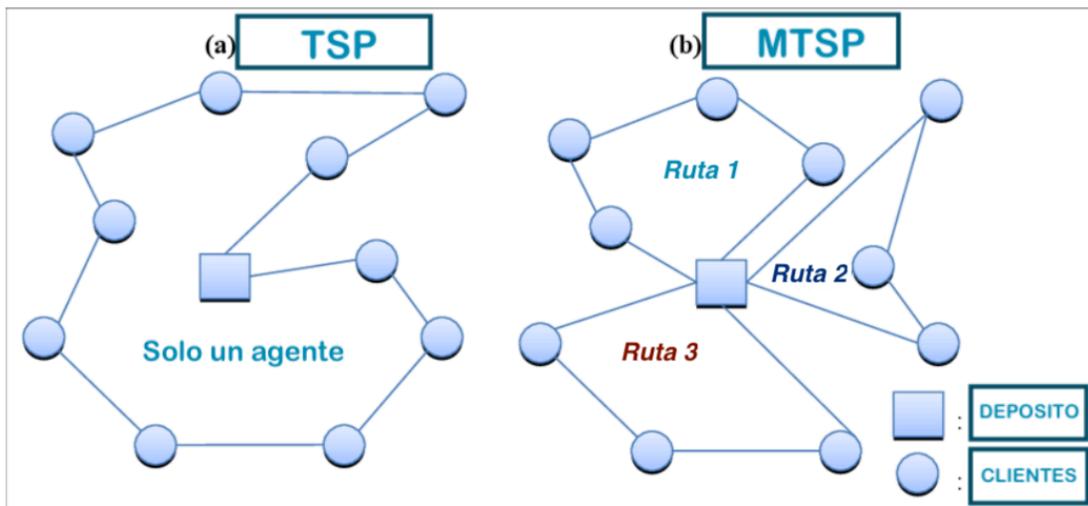


Figura 2.12. Diferencia entre TSP y MTSP (fuente propia)

El problema se define como, $G = (V, E)$ un grafo completo no dirigido ponderado y el número de recorridos por construir para un problema VRP. Donde V es el conjunto de nodos N (o ciudades a visitar), A es el conjunto de arcos y $C = (C_{ij})$ es la matriz de costo (distancia) asociada con cada arco $(i, j) \in A$. El costo de la matriz puede ser simétrica, asimétrica o euclidiana. Este último se refiere a la línea recta. Distancia

medida entre los dos nodos geográficamente dispersos. Hay m vendedores basados en el depósito, denotado como nodo o V_0 . El MTSP de depósito único consiste en encontrar recorridos para los vendedores de M sujetos a cada vendedor que comienza y termina en el depósito, cada nodo se encuentra exactamente en un recorrido, y el número de nodos visitados por un vendedor se encuentra dentro de un intervalo de tiempo (o distancia) predeterminado. El objetivo es minimizar el costo de las visitas de todos los nodos. Definimos la variable de decisión, X_{ij} , en la ecuación (2.9). Para cualquier vendedor, u_i , la ecuación (2.8) denota el número de nodos visitados en la ruta de ese vendedor hasta el nodo i , con los correspondientes parámetros K y L que indican el número mínimo y máximo de nodos visitados por cualquier un vendedor, respectivamente.

$$u_i \quad \text{Número de nodos visitados en cada ruta hasta el nodo } i \quad \forall i \in V \quad (2.8)$$

Por lo tanto, podemos indicar que $1 \leq u_i \leq L$ cuando $i \geq 2$, y cuando $X_{il} = 1$, luego $K \leq u_i \leq L$. La siguiente formulación del Programa Lineal Integral (ILP) es propuesto por (Kara, Laporte y Bektas, 2004). La introducción de múltiples agentes viajeros o vehículos del TSP básico introduce nuevas restricciones y modificación de la función objetivo.

El MTSP se suele formular utilizando una formulación de programación lineal de enteros de índice de doble asignación. Primero definimos la siguiente variable binaria:

$$X_{ijk} \quad \begin{array}{l} 1 \quad \text{Si la arista } (i, j) \text{ es usado por el} \\ \quad \quad \quad \text{vehículo } k \\ 0 \quad \text{De lo contrario} \end{array} \quad (2.9)$$

Entonces, un esquema general de la asignación basada en la formulación de programación lineal entera del MTSP se puede definir de la siguiente manera:

$$\min \sum_{(i,j) \in A}^n C_{ij} \sum_{k \in K}^n X_{ijk} \quad (2.10)$$

S.A

$$\sum_{j=2}^n X_{ij} = m \quad \text{Type equation here.} \quad (2.11)$$

$$\sum_{i=2}^n X_{ij1} = m \quad \text{Type equation here.} \quad (2.12)$$

$$u_i + X_{i1} < 1 \quad \forall i \in \{2, \dots, N\} \quad (2.13)$$

El objetivo de la ecuación (2.10) minimiza el costo total de viajar a todos los nodos, mientras que las restricciones (2.11) y (2.12) aseguran que a todos los M vendedores se les asignen rutas. Se imponen restricciones de grado (2.13). Las restricciones específicas de MTSP (2.13) se conocen como las restricciones de límites y (Kara y Bektas, 2004) introducen estos como las restricciones limitadas superior e inferior en el número de nodos visitados por cada vendedor. El valor de u_i se inicializa a 1 si y solo si el nodo i es el primer nodo en el recorrido de cualquier vendedor de desigualdad (2.10) prohíbe a un vendedor visitar solo un nodo en su recorrido. (2.11) y (2.12) aseguran que exactamente m vendedores salgan y regresen al nodo 1 (el depósito). Aunque las restricciones ya están implicadas, la formación de subtours entre todos los nodos en $V \setminus \{1\}$ (todos los nodos excepto el depósito) son eliminados por (2.13), ya que asegura que $u_i = u_i + 1$ si y solo si $X_{ij} = 1$.

Tabla 2.1. Procedimientos de solución propuestos para el MTSP.

Tipo de enfoque	Procedimiento de solución
Solución exacta	Formulaciones integrales de programación lineal Plano de corte Ramificación y corte Relajación LaGrange y ramificación y corte
Heurísticas	Heurística simple Algoritmos evolutivos Recocido simulado Búsqueda de tabú Algoritmos genéticos
Transformaciones	MTSP asimétrico a TSP asimétrico MTSP simétrico a TSP simétrico Múltiples depósitos MTSP a TSP

En la **Tabla 2.1** se muestran algunas de las soluciones más utilizadas para resolver este tipo de problemas. A continuación, consideramos una variante donde cada uno de los M vendedores de tiene una capacidad definida. Una analogía es tener vendedores que viajan en sus vehículos. Se tienen con coches espacio limitado, pero cada cliente visitado puede requerir un número diferente de demandas. Como una variante del MTSP, se le conoce como el problema del vendedor ambulante múltiple capacitado (CMTSP), pero en el contexto de esta tesis, se prefiere el nombre relacionado con el vehículo, problema de enrutamiento del vehículo (VRP).

El **MTSP** puede considerarse como una reducción del VRP, con las restricciones de capacidad eliminadas. Esto significa que todas las formulaciones y enfoques de solución propuestos para el **VRP** también son válidos y aplicables al **TSP**, al asignar capacidades suficientemente grandes a los vendedores (vehículos). Por otro lado, cuando sólo hay un vendedor único, entonces el **MTSP** se reduce a lo conocido **TSP**. Dado que el **TSP** es un caso especial del **MTSP**, todas las formulaciones y algoritmos descritos en este documento también son válidos para el problema anterior.

2.11.3. El problema de enrutamiento de vehículos (VRP)

Esencialmente, el VRP para la recolección trata con el mismo tipo de restricciones que en un problema de entrega al construir rutas de vehículos. Por lo tanto, este problema también intenta determinar la cantidad de vehículos necesarios para atender a los clientes, así como las rutas que minimizan la distancia total recorrida por los vehículos. Sin embargo, el vehículo para el problema de recolección está vacío cuando comienza desde el depósito, mientras que el vehículo para el problema de entrega comienza su ruta cargada con los bienes de los clientes que deben ser entregados. En el problema de recolección, los vehículos recolectarán bienes de un grupo de clientes y regresarán al depósito al final de la jornada laboral.

El VRP consiste en encontrar una colección de k circuitos simples, cada uno correspondiente a una ruta vehicular con costo mínimo, definida como la suma de los costes de los circuitos.

El problema se define como, $G = (V, E)$ un grafo completo no dirigido ponderado y el número de recorridos por construir para un problema VRP. En la **figura 2.13** se muestra un ejemplo de una solución del problema de enrutamiento de vehículos con 1 depósito, 25 clientes y 5 vehículos.

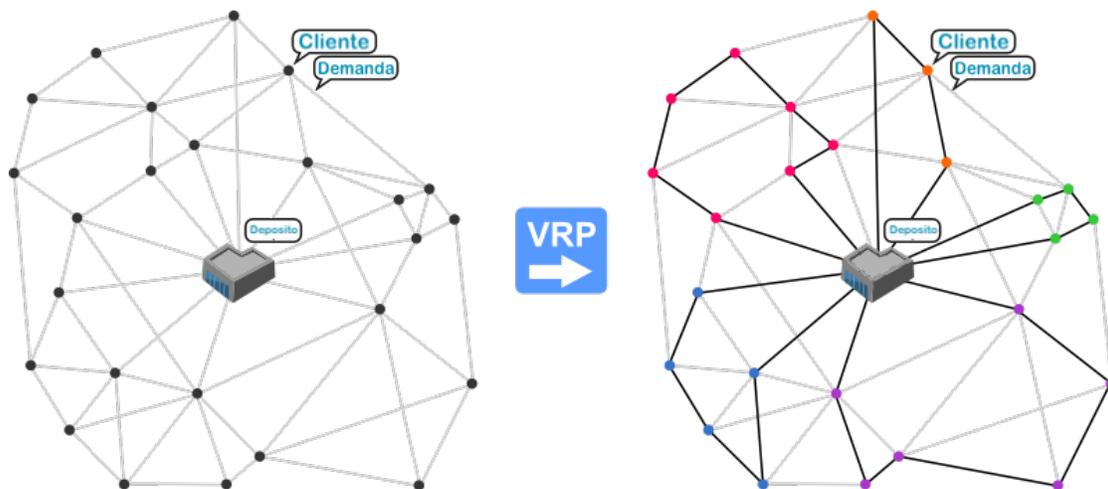


Figura 2.13. Ejemplo de una solución del problema de enrutamiento de vehículos con 1 depósito, 25 clientes y 5 vehículo (Lim & Wang, 2005)

El vértice V_d representa el depósito de donde parten k vehículos y los demás vértices son locales o clientes. Un costo no negativo, distancia o tiempo de viaje es dado por C_{ij} definido para cada arista del conjunto de aristas E . El VRP tiene como objetivo de fijar las rutas de los vehículos entre el depósito y los clientes que minimizan la distancia recorrida o el tiempo de recorrido. Las restricciones básicas del problema consisten en: cada sitio es visitado una sola vez por un solo vehículo; cada ruta se inicia en el almacén y se fija en el mismo depósito; todas las demandas de todos los clientes deben ser satisfechas.

En el VRP todos los clientes corresponden a entregas, las demandas son deterministas, conocidas y no pueden dividirse, los vehículos son idénticos y se basan en un único almacén central, solo se imponen las restricciones de capacidad para los vehículos, y el objetivo es minimizar el costo total (es decir, el número de rutas y / o su duración o tiempo de viaje) necesario para atender a todos los clientes. En general, el costo de viaje entre cada par de ubicaciones de clientes es el mismo en ambas direcciones, es decir, la matriz de costos resultante es simétrica, mientras que, en algunas aplicaciones, como la distribución urbana. Considere un conjunto de clientes representados por $V_c = \{1, n\}$, tal que cumplan con una demanda positiva la cual se asocia a cada cliente $i \in C$, para atender a estos clientes, tenemos que diseñar rutas para una flota con $K = \{1, \dots, k\}$ vehículos disponibles en un solo depósito. Cada ruta debe comenzar en el depósito, visita un subconjunto de clientes y luego vuelve al almacén. Todos los clientes deben ser visitados exactamente una vez. Cada vehículo tiene una capacidad máxima C , que limita el número de clientes que puede visitar antes de regresar al depósito. Para una mayor demostración, asumimos una flota homogénea de vehículos, pero la discusión presentada a continuación puede extenderse fácilmente a una flota heterogénea. En la **figura 2.14** se muestra un ejemplo de ubicaciones de un conjunto de clientes con sus respectivas coordenadas.

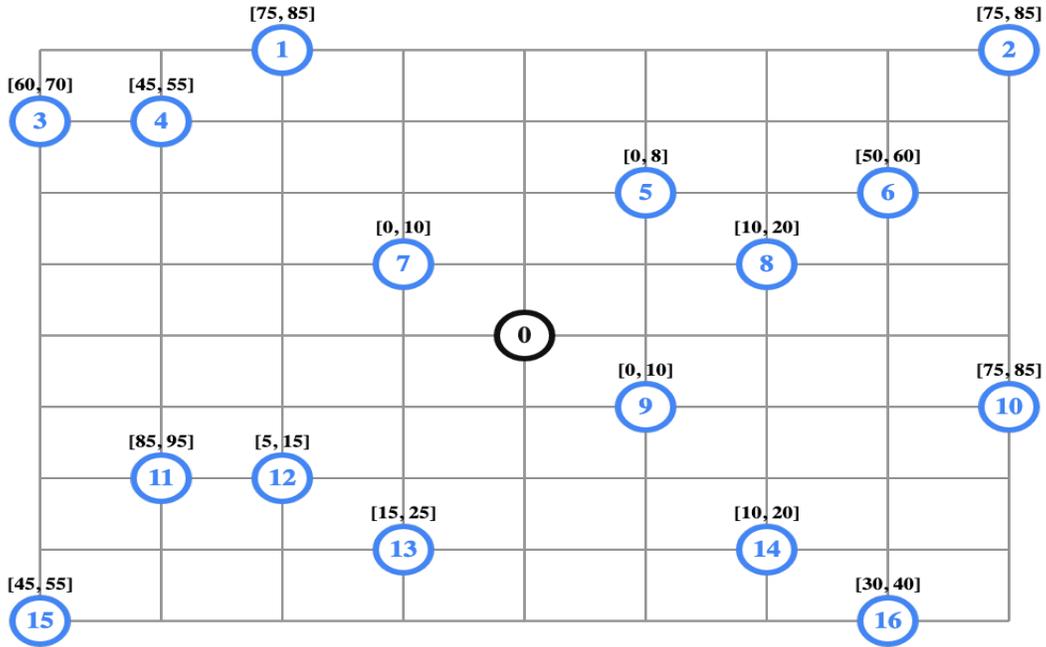


Figura 2.14. Coordenadas de clientes (Google Developers, 2020)

En la figura 2.15 se muestra la solución óptima para asignar clientes al conjunto de vehículos resolviendo el problema del enrutamiento de vehículos.

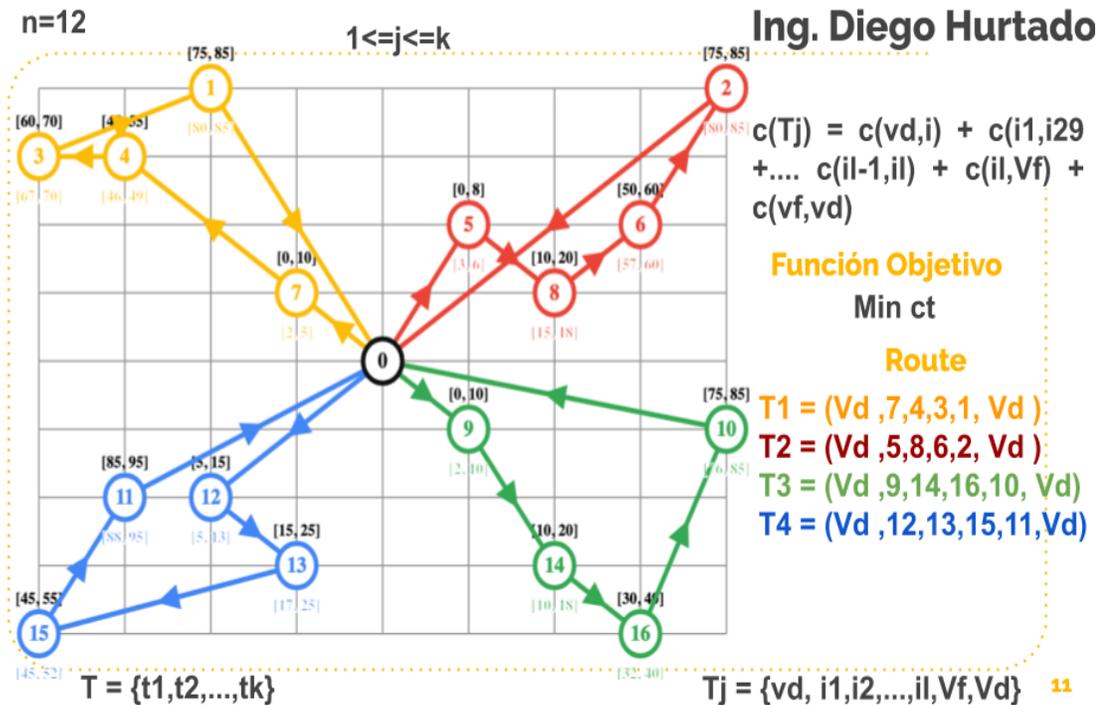


Figura 2.15. Asignación de recorridos óptimos (color) (Fuente propia)

En la **figura 2.16** se ilustra la asignación de vehículos.

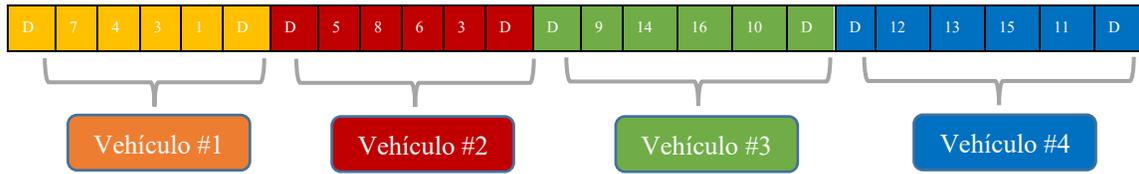


Figura 2.16. Codificación de asignación de rutas a vehículos.

En la figura anterior se ilustra la asignación de vehículos, donde se puede apreciar un conjunto de vehículos, los cuales cuenta con un conjunto de clientes asignados, cada cliente es atendido solo una vez y cada vehículo sale y regresa al depósito.

El objetivo de VRP es encontrar un conjunto de rutas con un costo mínimo, de tal manera que cada cliente es visitado exactamente una vez. Minimizar el costo total / distancia recorrida por todos los vehículos. Esta investigación se basa en los siguientes objetivos y restricciones.

Función Objetivo

$$\min \sum_{(i,j) \in A}^n C_{ij} \sum_{k \in K}^n X_{ijk} \tag{2.14}$$

S.A

$$\sum_{j \in V}^n X_{0jk} = 1 \quad \forall k \in K \tag{2.15}$$

$$\sum_{i \in V}^n X_{i0'k} = 1 \quad \forall k \in K \tag{2.16}$$

$$\sum_{i \in V_c}^n x_{ijk} = \sum_{i \in V_c}^n x_{jik} \quad \forall j \in V_c, k \in K \quad (2.17)$$

$$\sum_{i \in V}^n x_{ik}, n+1, k = 1 \quad \forall k \in K \quad (2.18)$$

$$\sum_{i \in V_c}^n d_i \sum_{k \in K}^n d_{ijk} \leq C \quad \forall k \in K \quad (2.19)$$

$$d_{ijk} \geq 0 \quad \forall k \in K, (i, j) \in A \quad (2.20)$$

$$X_{ijk} \in \{0,1\} \quad \forall (i,j) \in A, k \in K \quad (2.21)$$

La ecuación (2.14) minimiza el costo total de viaje a seguir por cada vehículo k , bajo las siguientes restricciones. El costo total se puede interpretar como el recorrido total en unidades distancia, tiempo o costo. La ecuación (2.15) proporciona el conjunto de restricciones de flujo que aseguran que el número de clientes j son directamente conectados desde el depósito de origen al iniciar el recorrido utilizando el vehículo k , sea igual a uno. Partiendo del depósito solo un nodo destino puede ser atendido. El vehículo inicia la ruta desde el depósito. El vehículo vuelve al depósito al final de la ruta. La ecuación (2.16) proporciona el conjunto de restricciones de flujo que aseguran que el vehículo vuelve al depósito final de la ruta. La ecuación (2.17) proporciona las restricciones que el flujo de entrada y salida deben ser iguales, excepto los nodos de depósito. La ecuación (2.18) proporciona el conjunto de restricciones que garantiza que para cada vehículo k solo debe existir un nodo que conecte directamente con el depósito $n+1$. La restricción (2.19) garantiza que el conjunto de restricciones representadas en las desigualdades es la suma de las demandas q_i de cada nodo en el recorrido x , que no

debe exceder la capacidad C total del vehículo k . Donde N indica el número de clientes a atender, k indica la flotilla de vehículos con capacidad C . La restricción (2.20) garantiza que el modelo lineal se defina como un modelo lineal entero de variables binarias. La restricción (2.21) garantiza que el modelo lineal se defina como un modelo lineal de no negatividad tomando un modelo lineal entero de variables binarias.

2.12. Variantes del VRP

El propósito de esta sección es clasificar las variantes en la literatura sobre el VRP, en base a los detalles característicos del problema de VRP estudiado. A continuación, se presentan revisiones de la literatura existente enfocadas sobre variantes o aspectos específicos del VRP. El VRP capacitado (Laporte, 2009), VRP periódicos (Campbell & Wilson, 2014), el VRP con ventanas de tiempo (Bräysy & Gendreau, 2005), VRPs dinámicos (Pillac, Gendreau, Guéret, y Medaglia, 2013), pickup y problemas de entrega (Berbeglia, Cordeau, Gribkovskaia, y Laporte, 2007), enrutamiento de vehículos con depósitos múltiples (Montoya, López, Nieto, Felizzola y Herazo, 2015), enrutamiento de vehículos con entregas divididas (Archetti y Speranza, 2012), entre otras.

Esta revisión a menudo se centra en revisar los métodos de solución propuestos, en lugar de identificar diferencias en supuestos y características entre variantes de VRP. Por ejemplo, la bibliografía categorizada sobre enfoques metaheurísticos para diferentes variantes de VRP está disponible en (Gendreau, Potvin, Bräumlaysy, Hasle y Lokketangen, 2008).

El VRP básico hace una serie de suposiciones, incluida la utilización de una flota homogénea, un depósito único, una ruta por vehículo, etc. Estas suposiciones pueden eliminarse mediante la introducción de restricciones adicionales al problema. Esto implica aumentar la complejidad del problema y, por restricción, clasifica el problema extendido como un problema np-hard (Lageweg, Lawler, Lenstra & Rinnooy, 1981). La mayoría de estas restricciones adicionales a menudo se implementan de forma

aislada, sin integración, debido a la mayor complejidad de la solución de tales problemas. En las siguientes secciones, estas variantes se presentan de forma aislada, antes de proponer una formulación integrada para el problema en cuestión.

Los modelos actuales de VRP, sin embargo, son inmensamente diferentes de los introducidos por (Dantzig & Ramser, 1959), ya que apuntan cada vez más a incorporarlos a la vida real. Complejidades, como por ejemplo los tiempos de viaje dependientes del tiempo (que reflejan la congestión del tráfico), las ventanas de tiempo para recogida y entrega, e información de entrada (por ejemplo, información de demanda) que cambia dinámicamente a lo largo del tiempo.

Introducimos definiciones generales para las variantes de VRP destacadas en nuestra revisión de la literatura y discutir una serie de temas que mejor se enfocan a nuestro problema de investigación en la **figura 2.17**, es decir, el problema de enrutamiento del vehículo de recolección de residuos con Windows de tiempo (WCVRPTW).

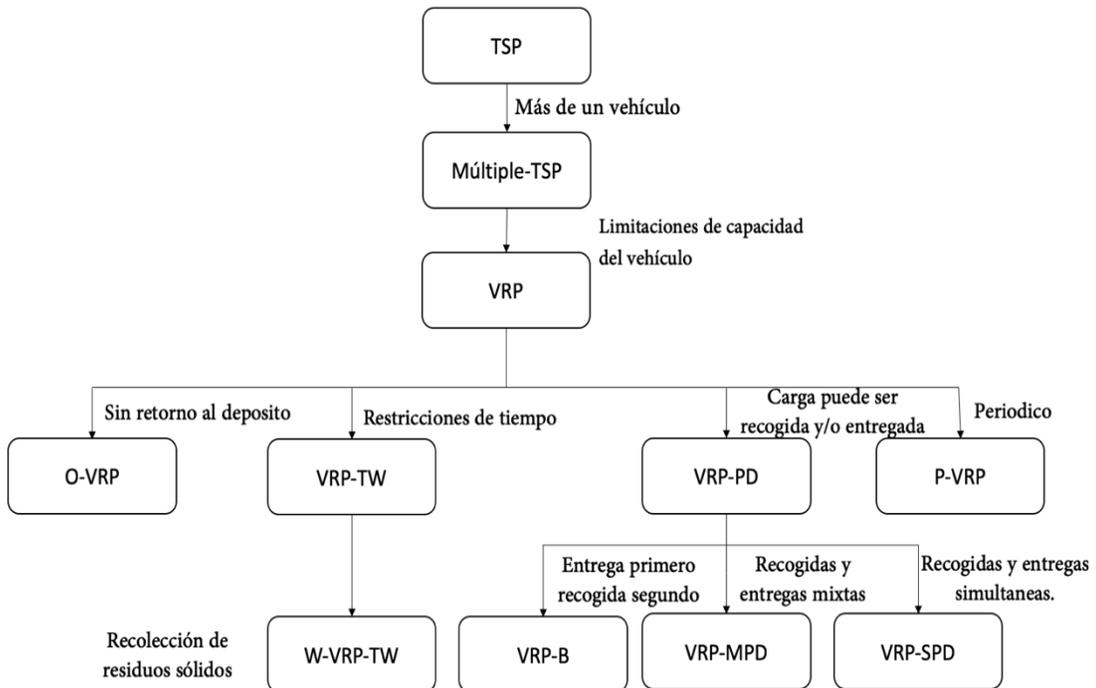


Figura 2.17. Variantes VRP basado en (Irnich, Schneider & Vigo, 2014)

Este VRP clásico se ha ampliado de muchas maneras al introducir aspectos adicionales de la vida real o características, lo que resulta en un gran número de variantes de la VRP.

2.12.1.VRPPD

En el VRP con recolección y entrega (VRPPD, P-Pickup, D-Delivery), los bienes deben ser recogidos de una determinada ubicación y dejados en su destino. La recolección y la devolución deben hacerse por el mismo vehículo, por lo que el lugar de recogida y el lugar de entrega deben incluirse en la misma ruta (Tasan & Gen, 2012).

2.12.2.VRPB

Un problema relacionado es el VRP con backhauls (VRPB), donde un vehículo de entregas y recogidas en una ruta (Pradenas, Oportus y Parada, 2013). El término de “backhauls” se traduciría como el viaje de vuelta, se trata de una idea de negocio el cual se basa en evitar que el vehículo viaje vacío en el viaje de vuelta. Algunos clientes requieren entregas (a las que se hace referencia como linehauls) y otras requieren recolecciones (a las que se hace referencia como backhauls). La combinación de linehauls y backhauls ha demostrado ser muy valiosa para la industria de logística de cadena de suministros. Así, se planifica una ruta en base a los proveedores que deseen aprovechar el espacio de vuelta para su carga. Los costos de transporte y la distancia total recorrida pueden reducirse significativamente y la carga del vehículo. Entre algunos de los beneficios claves son el ahorro de combustibles, ahorro de recursos, ahorro de tiempo y reducción de impacto ambiental.

2.12.3.MDVRP

El Multi-Depot VRP (MDVRP) asume que los depósitos múltiples se distribuyen geográficamente entre los clientes (Montoya-Torres et al., 2015). El VRP periódico (PVRP) se usa cuando la planificación se realiza durante un período determinado y las entregas al cliente se pueden realizar en diferentes días (Gulczynski, Golden, & Wasil,

2011). Para el PVRP, los clientes pueden ser visitados más de una vez, aunque a menudo con una frecuencia limitada.

En los siguientes párrafos, detallamos más variantes específicas del VRP que modeló aspectos de la vida real y que comenzó a recibir más atención de investigación en los últimos años: el Open VRP, el VRP dinámico y el VRP con ventanas de tiempo.

2.12.4.OVRP

En el open VRP (OVRP) no se requiere que los vehículos regresen al depósito central después de visitar el último cliente si regresan, deben visitar a los mismos clientes en orden inverso. Adicionalmente, el OVRP a menudo tiene dos objetivos de optimización: minimizar el número de vehículos utilizados y (dado este número de vehículos) minimizando la distancia total (o, a veces, el tiempo) recorrida. En la práctica, el OVRP ocurre cuando la flota de vehículos no es propiedad de la propia compañía o cuando la flota de vehículos disponible es incapaz de satisfacer la demanda de sus clientes, de manera que (parte de) las actividades de distribución son contratadas a un proveedor externo de logística (Repoussis, Tarantilis, Bräysy, & Ioannou, 2010). La solución OVRP luego indica la cantidad de vehículos que se necesita. Además, el OVRP podría ser utilizado en el caso de recogida y entrega, cuando después de entregar las mercancías a determinados clientes, los vehículos recogen productos de los mismos clientes, pero en orden inverso (Salari, Toth y Tramontani, 2010).

2.12.5.DVRP

En el Problema de enrutamiento dinámico de vehículos (DVRP), a veces referido como problema de enrutamiento vehicular en línea. Los problemas de enrutamiento han surgido recientemente gracias a los avances en las tecnologías de comunicación e información que permiten obtener y procesar información en tiempo real. En este caso, algunas de las órdenes se conocen de antemano antes del comienzo del día laborable, pero a medida que avanza el día, llegan nuevos pedidos y el sistema tiene que

incorporarlos a las rutas. El sistema de comunicación entre los conductores y el sistema donde los tours son calculados, se asume la sede de la empresa y los conductores parten. El encargado puede comunicar periódicamente a los conductores las nuevas visitas que se les asignen. De este modo, durante el día, cada conductor siempre tiene un conocimiento sobre los próximos clientes asignados a él.

Adicionalmente, a partir de la experiencia de autores, no hay instancias de referencia disponibles para probar y comparar los métodos de solución propuestos de forma objetiva.

2.12.6. VRPTW

El problema de enrutamiento del vehículo con ventanas de tiempo (VRPTW) es una de las variantes del VRP, aquí se asocia una ventana de tiempo con el cliente i . Cada cliente i debe ser atendido dentro de ese período de tiempo $[a_i, b_i]$. Si un vehículo k arriba anteriormente de la hora de inicio a_i , entonces tiene que esperar hasta que sea posible el inicio del tiempo de servicio s_i , pero un vehículo k no puede venir después del tiempo final b_i . Si un vehículo k en el intervalo de la ventana de tiempo, el vehículo k suministra la demanda en el momento de la llegada y finalmente si el vehículo k llega con posterioridad a b_i entonces el cliente no atenderá al chofer del vehículo k . En el modelo de ventana de tiempo suave, un vehículo puede dar servicio a un cliente más allá de la restricción de la ventana de tiempo con una penalización.

Por lo general, se expresa como un marco de tiempo para cada cliente. Normalmente, si el vehículo llega temprano a un cliente, debe esperar hasta que sea posible el inicio del servicio. El VRPTW es uno de los problemas más estudiados en el campo de la optimización combinatoria. Es una variante del problema de enrutamiento del vehículo (VRP) definido por (Dantzig & Ramser, 1959). En el VRP con ventanas de tiempo (VRPTW) la restricción de ventana de tiempo limita el tiempo en que un cliente está disponible para ser atendido por un vehículo.

Esta restricción por lo general se expresa como un marco de tiempo para cada cliente. Normalmente, si el vehículo llega antes de la apertura de la ventana del tiempo de un cliente, debe esperar hasta que sea posible el inicio del servicio. Una solución para el VRPTW es un conjunto de rutas que consiste en una secuencia de visitas a los clientes, donde cada ruta se asigna a un vehículo y todos los clientes se visitan dentro de sus ventanas de tiempo. El volumen total asignado a cada ruta no debe exceder la capacidad del vehículo. El desafío es encontrar una solución que minimice el número total de vehículos utilizados y la distancia recorrida.

El problema del CVRPTW es representado por un grafo completo no dirigido ponderado $G = (V, E)$, con un conjunto de vértices $V = \{0, 1, 2, \dots, n\}$ y un conjunto de aristas E .

Consiste en un conjunto de clientes V_d , con todos los pesos positivos; El nodo $V_d \in V$ representan el depósito, es decir, del cual todos k vehículos parten los recorridos pasando por cada vértice exactamente una vez, y finalmente regresando al depósito, a esto se le llama tour. La longitud de un tour es la suma de las longitudes de las aristas en el camino que define el recorrido. Ningún arco termina en el nodo V_d del depósito y todas las rutas comienzan y terminan en el depósito V_d . El conjunto de vértices representado por los clientes $V_d \in V$ donde $V \setminus V_d$ y $V_c = \{V_1, \dots, V_n\}$; y los demás vértices son clientes V_c ; El conjunto de n vértices que denotan a los clientes se denota como n .

El conjunto de arcos A denota todas las conexiones posibles entre los nodos (incluido el nodo que denota el depósito). E representa el conjunto de las parejas (i, j) : donde i, j pertenece $\in V$ al conjunto de los vértices se le denomina el conjunto de aristas. $E = \{(V_i, V_j) : i \neq j\}$ es el conjunto de arcos, cada uno con un costo de viaje no negativo C_{ij} . Asociamos un costo C_{ij} asociado a la distancia o tiempo de viaje definido para cada arista del conjunto de aristas y un tiempo t_{ij} con cada arco $(i, j) \in A$ de la red de enrutamiento. El tiempo de viaje $t_{i, j}$ puede incluir el tiempo de servicio s_i en el cliente i . Cada cliente $V_c \in V \setminus \{V_d\}$ tiene una demanda no negativa q_i , $i \in C$; Cada ruta tiene un límite diario de carga máxima de recolección de basura permita por cada cliente V_c a la

cual debe ser satisfecha por un vehículo y la capacidad C del vehículo debe respetarse; Una flota homogénea con cada vehículo con capacidad Q , es disponible en el depósito V_d .

t_{ij} y C_{ij} representan el tiempo de viaje y el costo asociado con el arco (i, j) . Respectivamente cada nodo $i \in V_c$ tiene un tiempo de servicio asociado s_i y una ventana de tiempo $[a_i, b_i]$ y definimos el arco q_i como la cantidad de recolección de un cliente $i \in V_c$. Un vehículo puede llegar antes del comienzo de la ventana de tiempo (es decir, a_i), lo que significa que incurre en tiempo de espera hasta que el servicio sea posible. Sin embargo, ningún vehículo puede llegar más allá del cierre de un intervalo de tiempo dado, b_i . Los vehículos también deben salir del depósito dentro de la ventana de tiempo del depósito $[a_0, b_0]$ y deben regresar antes o en el momento $b_n + 1$. Suponiendo que el tiempo de espera se permite sin costo, podemos suponer que $a_0 = b_0 = 0$; es decir, todas las rutas comienzan en el tiempo 0.

Para modelar el problema del depósito, se divide en un depósito inicial y final $\{0,0\}$. El problema puede luego ser modelado usando diferentes tipos de variables: $X_{ijk} \in \{0,1\}$ es uno si y sólo si el vehículo l de K usa el arco $(i, j) \in E$, q_i representa la demanda acumulada en el nodo $i \in V_c$ para el vehículo $l \in K$, y w_i representa la hora de inicio del servicio en el nodo $i \in V_c$ para el vehículo $l \in K$. $W_{ijk} \in V$, $k \in K$, especificando el inicio del servicio en el nodo i cuando es reparado por un vehículo k .

La meta del CVRPTW es generar un número de recorridos de los vehículos entre el depósito y los clientes para satisfacer la demanda de los clientes tal que la distancia total sea la mínima cumpliendo con las restricciones para construir un problema CVRPTW. Las restricciones básicas del problema consisten en: cada sitio es visitado una sola vez por un solo vehículo; cada ruta se inicia en el mismo depósito; todas las demandas de todos los clientes deben ser satisfechas. El vehículo no debe llegar antes de tiempo y no más tarde de tiempo. Un vehículo que llega antes de tiempo. Producirá un tiempo de espera. Un vehículo que llegue después de la hora incurrirá en un retraso.

Todos los vehículos programados tienen que regresar al depósito dentro del tiempo máximo de operación para cada vehículo.

Una ventana de tiempo puede describirse como una ventana de oportunidad para entregas. Es una extensión del VRP que se ha investigado exhaustivamente (Ibaraki et al., 2005; Taillard, & Agazzi, 1999; Taillard, & Gambardella, 1997).

Una ventana de tiempo es el período de tiempo durante el cual las entregas se pueden hacer a un cliente específico i , y tiene tres características principales:

- El tiempo de llegada permitido, a_i , también conocido como el tiempo de apertura de la ventana de tiempo
- Última hora de llegada permitida, b_i , también conocido como la hora de cierre
- Si la ventana de tiempo se considera suave o dura

Considere el ejemplo ilustrado en la **figura 2.18**, donde el cliente i solicita la entrega entre 07:30 y 17:00. Para distinguir entre el tiempo real y el tiempo de llegada especificado.

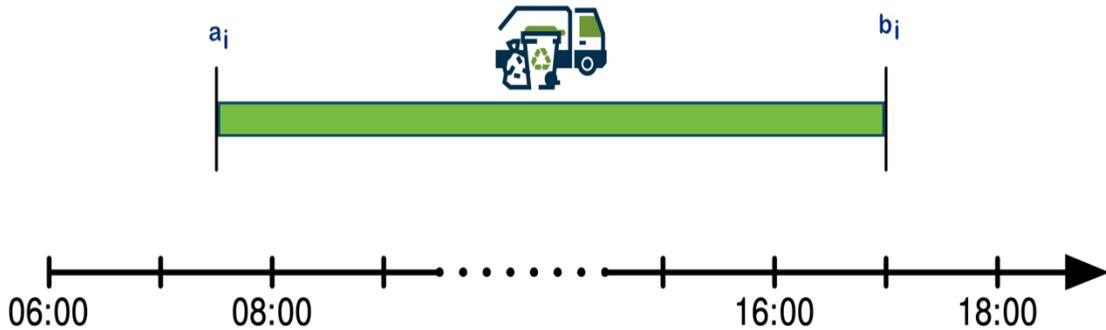


Figura 2.18. Ventana de tiempo de vehículo

La variable a_i denota el tiempo real de llegada al nodo i . En caso de que la hora de llegada real en el nodo i , denotado por a_i , sea más temprano que la primera llegada permitida al nodo, e_i , entonces el vehículo incurrirá en un tiempo de espera, w_i , que se puede calcular como $w_i = \max. \{0, e_i - a_i \}$.

La introducción de ventanas de tiempo al VRP básico requiere la introducción de tres nuevas restricciones.

$$a_0 = w_0 = s_0 = 0 \quad (2.22)$$

La restricción (2.22) garantiza que $a_0 = b_0 = 0$; es decir, todas las rutas comienzan en el tiempo 0.

$$\sum_{k=1}^n \sum_{i=0}^n x_{ijk} (a_i + w_i + s_i + t_{ij}) < a_j \quad \forall j \in \{1, 2, \dots, N\} \quad (2.23)$$

La ecuación (2.23) proporciona el conjunto de restricciones que garantiza la atención del nodo i antes del nodo j . Considerando que la suma del tiempo de inicio de servicio al nodo i , el tiempo de servicio asignado al nodo i , el tiempo de recorrido del nodo i al nodo j y el tiempo de inicio de servicio del nodo j , debe ser menor o igual a $(1 - X_{i0k})$.

$$e_i < (a_i + w_i) < l_i \quad \forall i \in \{1, 2, \dots, N\} \quad (2.24)$$

La restricción (2.24) garantiza que se respete el conjunto de restricciones que indiquen el tiempo de inicio de servicio W_{ik} al nodo i no debe exceder los límites de la ventana de tiempo del nodo i definido por el intervalo $[a_i, b_i]$.

2.12.7. WCVRPTW

El problema de la recolección de desechos consiste en el enrutamiento de los vehículos para recolectar los desperdicios de los clientes dentro de un período de tiempo determinado, al tiempo que minimiza los costos de viaje. Este problema se conoce como el problema de enrutamiento del vehículo de recolección de residuos con ventanas de tiempo (WCVRPTW). WCVRPTW se diferencia del VRPTW tradicional en que los

vehículos de recolección de desechos deben vaciar su carga en los sitios de disposición. Los vehículos deben estar vacíos al regresar al depósito. Se permiten múltiples viajes a sitios de disposición para los vehículos. El problema se ilustra en la **figura 2.19** para un solo vehículo y múltiples sitios de eliminación.

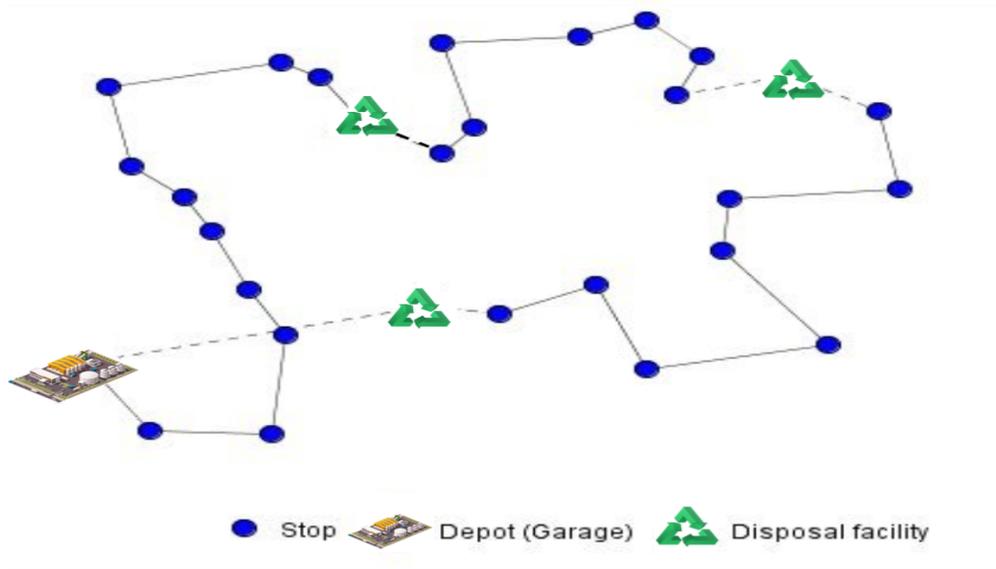


Figura 2.19. Esquema de WCVRPTW (Fuente propia)

Kim et al. (año) definieron informalmente el WCVRPTW como sea $G = (V, A)$, donde el conjunto de nodos $V = V_d \ \& \ V_f \ \& \ V_c$ el cual consiste en un depósito $V_d = \{0\}$, m sitios de disposición $V_f = \{1, \dots, m\}$, n clientes $V_c = \{m+1, 1, \dots, m+2\}$ y el conjunto de arcos $A = \{(i, j) \mid i, j \in V, i \neq j\}$. Sea $K = \{1, \dots, k\}$ el conjunto de vehículos y sea t_{ij} y C_{ij} el tiempo de viaje y el costo asociado con el arco (i, j) , respectivamente.

Para modelar la pausa del almuerzo se definen dos nuevas variables. Una nueva variable binaria y_{ijk} indica si el almuerzo fue tomado entre los nodos i y j visitados por el vehículo k . Una variable continua r_l da la proporción de conducción realizada entre la parada anterior y la siguiente cuando el vehículo k tomo un descanso para almorzar. La ecuación que garantiza que la duración del almuerzo sea tenida en cuenta. En la **figura 2.20** se muestra una lustración al problema de enrutamiento del vehículo de recolección de residuos con ventanas de tiempo (WCVRPTW).

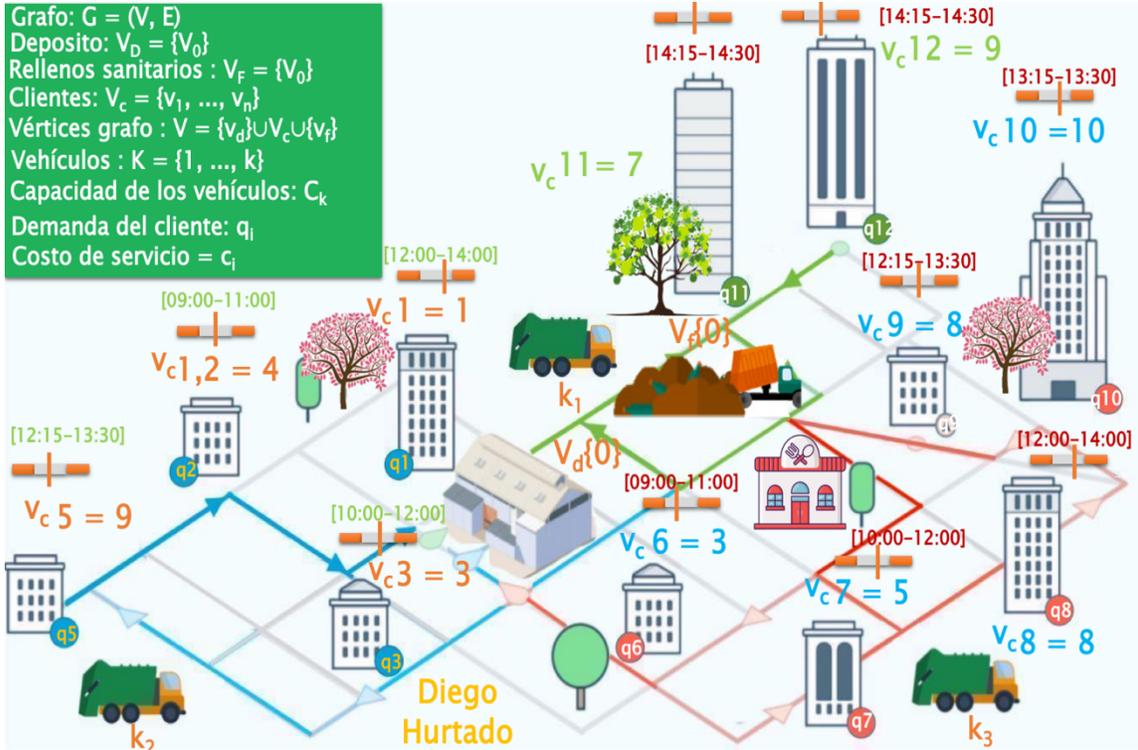


Figura 2.20. WCVRPTW en una Smart City (fuente propia)

$$\sum_{i \in \{0,0\}}^n d_{ik} = 0 \quad \forall j \in V_c, k \in k \quad (2.25)$$

$$\sum_{(i,j) \in A}^n Y_{ijk} = 1 \quad \forall k \in K \quad (2.26)$$

$$Y_{ijk} \leq X_{ijk} \quad \forall (i,j) \in A, k \in K \quad (2.27)$$

$$w_{ik} + y_i + y_{ijksu} + t_{ij} \leq (1 - x_{jik}) M \quad \forall k \in K, (i,j) \in A \quad (2.28)$$

La restricción (2.25) garantiza que los vehículos deben estar vacíos al inicio de las rutas y al final de las rutas cuando regresan al depósito. La restricción (2.26) garantiza que cada vehículo debe tomar exactamente un descanso para almorzar. La restricción

(2.27) garantiza que el almuerzo solo pueda ser entre dos nodos i y j si solo si están conectados. La restricción (2.28) garantiza que el almuerzo se lleva a cabo dentro de su horario de servicio en su ventana de tiempo.

Los resultados indican que la literatura de enrutamiento del vehículo consta de una amplia gama de variantes de problemas. Los investigadores prestan cada vez más atención a las variantes de VRP que incluyen características de la vida real y supuestos, lo que hace que sus modelos sean más realistas y su solución y enfoque sea más aplicable en la práctica. Sin embargo, las características de la vida real a menudo se consideran individualmente con un número limitado de otras características. Por lo tanto, la investigación se centra en el problema de investigación para la recolección de residuos sólidos y sus métodos de solución para resolver este problema.

2.13. Tabla de restricciones de los tipos de problemas de ruteo

Esta sección presenta las restricciones para cada adaptación del problema de ruteo de vehículos desde su primera versión del problema del agente viajero que solo consideraba un agente viajero hasta la versión del VRP con ventanas de tiempo, con un conjunto de vehículos y una capacidad limitada, mostrándose en cada etapa las restricciones adicionales.

Tabla 2.2. Tabla de restricciones de los modelos de enrutamiento de vehículos.

Variación	Restricciones agregadas
<p>TSP (Dantzig, Fulkerson & Johnson, 1954)</p>	<p>El agente inicia la ruta desde un cliente y vuelve al mismo cliente. Cada cliente $V_c \in V \setminus \{V_0\}$ es visitado exactamente una vez Costo asociado de viaje entre el cliente i y j (i,j) tiene que ser mayor a 0</p>
<p>MTSP (Dantzig & Ramser, 1959)</p>	<p>Cada k agente o vehículo inicia la ruta desde el depósito y vuelve al depósito. Cada k agente o vehículo tiene un número mínimo y máximo de clientes por visitar</p>

(Dantzig, G. B., & Ramser, J. (1959)	La demanda acumulada de recolección de todos los clientes no exceda la capacidad del vehículo no debe superarse.
VRPTW (Solomon, 1984)	Garantizar la factibilidad de las rutas con respecto a las restricciones de ventana de tiempo. No se puede visitar al cliente antes de la apertura de la ventana de tiempo a_i
WCVRPTW (Kim et al., 2006)	El vehículo debe partir y volver vacío al depósito. Garantizar la factibilidad de las rutas con respecto a las restricciones de ventana de tiempo de las horas de comida de los conductores Cada vehículo $k \in K \in V \setminus \{V_0\}$ puede tomar un break exactamente una vez
Tesis (2020)	Cada vehículo inicia la ruta desde el depósito y vuelve al depósito. Cada cliente $V_c \in V \setminus \{V_0\}$ es visitado exactamente una vez Costo asociado de viaje entre el cliente i y j (i,j) tiene que ser mayor a 0 La demanda acumulada de recolección de todos los clientes no exceda la capacidad del vehículo No debe superarse la cantidad máxima de basura por cliente que se puede recoger. Garantizar la factibilidad de las rutas con respecto a las restricciones de ventana de tiempo

En la **Tabla 2.3** se muestra las restricciones adicionales para cada modelo del problema de ruteo vehicular.

2.14. Diferentes métodos de resolución del VRPTW

En cuanto a los métodos para la resolución del problema del ruteo vehicular, en este apartado se habla acerca de las técnicas adoptadas para resolver este problema, algoritmos poblacionales, búsqueda local, métodos exactos y técnicas de aprendizaje máquina para la optimización del problema, se mencionan, primeramente, algunos

conceptos de cada categoría y finalmente se analizan las representaciones adoptadas para acotar este tipo de problemática para tener un diseño de cómo se va realizar el proyecto. En la **figura 2.21** se ilustran los métodos utilizados para resolver el VRPTW, heurísticas, aprendizaje máquina, metaheurísticas, construcción, búsqueda local, redes neuronales, basadas en poblaciones, basadas en trayectorias y métodos constructivos.

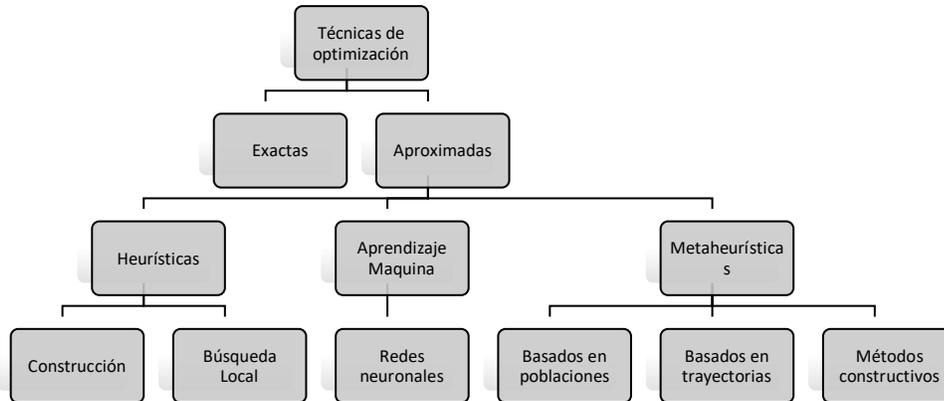


Figura 2.21. Métodos para resolver el VRPTW (fuente propia)

La búsqueda local comienza a partir de la solución inicial (por ejemplo, proporcionada por una heurística constructiva de solución inicial), y posteriormente se mueve de la solución presente a una solución vecina en el espacio de búsqueda donde cada solución tiene un número relativamente pequeño de soluciones factibles. Las soluciones de los vecinos y cada uno de los movimientos están determinados por los operadores del barrido. Los operadores del primer grupo mueven uno o más clientes de una posición en la ruta a otra posición en la misma ruta y son llamados operadores de ruta.

Los algoritmos poblacionales se basan en la evolución de un aprendizaje colectivo dentro de una población de individuos, cada individuo dentro de esta población figura como una solución dentro del espacio de búsqueda a un problema determinado. Métodos que son poblacionales, lo que significa que trabajan con varias soluciones a la vez, y no con una, como lo hacen la mayoría de las otras heurísticas (por ejemplo, el recocido simulado), con lo que evitan quedar atrapados en óptimos locales. La población es arbitrariamente inicializada y la evolución va dirigida hacia los mejores individuos, para mejorar las regiones del espacio de búsqueda se aplican procesos

aleatorios. Se trata de desarrollar algoritmos para solucionar optimización, búsqueda, bajo el esquema de la “supervivencia del más apto” que postuló Charles Darwin en su teoría de la evolución de las especies. Estos algoritmos conforman la computación evolutiva (Yao, Liu & Lin, 1999).

Algoritmo 1 Programación evolutiva (Yao, Liu & Lin, 1999)

Generar la población inicial

Evaluar la población inicial

Repetir:

 Aplicar operador de mutación

 Aplicar operador de cruzamiento

 Evaluar cada hijo

 Realizar la selección

Mientras: no se cumpla la condición de paro.

El algoritmo de la programación evolutiva es el mostrado en el Algoritmo 1. La población inicial es de tamaño n .

Uno de los algoritmos poblacionales más utilizados son los algoritmos genéticos (AG), los AG son una técnica heurística de búsqueda u optimización de soluciones, originalmente motivada por el principio darwiniano de evolución a través de la selección (genética). Un AG utiliza una versión muy abstracta de los procesos evolutivos para desarrollar soluciones a problemas dados. Cada AG opera en una población de cromosomas artificiales. Estas son cadenas en un alfabeto finito (generalmente binario). Cada cromosoma representa una solución a un problema y tiene afinidad, un número real que es una medida de cuán buena es una solución para el problema particular. Fueron propuestos por (Holland, 1975) con el objetivo de resolver problemas de optimización combinatoria.

Algoritmo 1.1 Algoritmo genético (Holland, 1975)

Generar la población inicial

Evaluar la población inicial

Repetir:

Realizar la selección

Aplicar operador de cruzamiento

Aplicar operador de mutación

Evaluar cada hijo

Mientras no se cumpla la condición de paro

El algoritmo genético básico propuesto por (Holland, 1975), se puede apreciar cada uno de los componentes.

La posibilidad de descubrir nuevos métodos para resolver problemas de optimización combinatoria, sin ningún razonamiento diseñado para métodos tradicionalmente utilizados, hace que los algoritmos basados en aprendizaje máquina como redes neuronales sean una nueva opción atractiva que tiene el potencial de ser un importante hito en el camino hacia la solución de estos problemas. Sin embargo, los primeros intentos de aplicar redes neuronales para resolver el VRP han sido bastante inútiles. Un enfoque ingenuo sería entrenar un modelo para una instancia específica considerando cada instancia por separado. Con este enfoque, un algoritmo necesita tomar muchos datos para entrenar un modelo, tal vez millones de ellas. Obviamente, esto el enfoque no es práctico ya que el método debe ser comparable a los algoritmos existentes no sólo en términos de calidad de la solución, pero también en términos de tiempo de ejecución.

2.15. Trabajo relacionado de WCVRPTW

El uso de computadoras para resolver problemas de recolección de residuos ha sido reportado por varios investigadores durante los últimos 29 años. Desde sus primeras aplicaciones por (Ong, Goh, Poh & Lim, 1990) con un enfoque para optimizar la recolección de residuos en Singapur, se desarrolló una ruta heurística con un enfoque de ruta del primer grupo. La prueba se ejecuta utilizando datos de uno de los distritos. El VRP exige una cantidad considerable de tiempo en la recopilación de datos, y la mayor parte del esfuerzo de programación ha sido en el manejo de datos. El esfuerzo de enrutamiento real constituye una parte relativamente pequeña del sistema. Las pruebas han demostrado que, con el VRP, hay un ahorro constante en la distancia recorrida por los vehículos. Se puede lograr, lo que se traduce en ahorros en combustibles y trabajos de mantenimiento de vehículos y eventualmente en costos operacionales.

Una de las primeras soluciones propuestas para la recolección de residuos fue por (Kulcar, 1996) utilizando una metodología que combina un método de investigación de operaciones con la ingeniería de sistemas. En este trabajo se ilustra cómo son los costos de transporte de residuos minimizados en un área urbana importante. Por la complejidad de la situación de tamaño real, se implementa un modelo para considerar un conjunto de puntos para las rutas de recolección en lugar de los arcos que forman las calles. Varios medios de transporte incluyendo transporte en vehículo, ferrocarril y canal son evaluados. Esto muestra cómo una formulación simple de un problema como un programa lineal proporciona a los responsables de la toma de decisiones soluciones eficientes en el complejo problema de la administración urbana.

(Tung & Pinnoi, 2000) consideraron solo un sitio de disposición y formularon el problema en un programa de enteros mixtos, donde modifican el algoritmo de inserción de Solomon y lo aplican a un problema de recolección de desechos sólidos en Hanói, Vietnam. Proponen un procedimiento heurístico para resolver un VRSP de recolección de residuos en Hanói. El problema se caracteriza por una carga y descarga VRSP con ventanas de tiempo, tiempo de llegada y restricciones de tiempo total. La heurística propuesta consta de dos fases: la Fases de construcción y mejora. En comparación con la

práctica actual, la heurística propuesta muestra un buen desempeño con respecto de la calidad de la solución y el tiempo computacional.

El problema de la recolección de residuos consiste en enrutar los vehículos para recolectar los desechos de los clientes dentro de su ventana de tiempo determinada minimizando los gastos de viaje. Este problema se conoce como el problema de enrutamiento del vehículo de recolección de residuos con ventanas de tiempo (WCVRPTW).

En este enfoque, los autores consideran restricciones como ventanas de tiempo, varios viajes a los rellenos sanitarios y una pausa de almuerzo para los conductores de los vehículos. Los objetivos del problema son minimizar el tiempo de viaje, la reducción del número de rutas y el equilibrio de la carga de trabajo.

(Teixeira et al, 2004) describen un estudio de planificación de rutas de vehículos para la recolección de residuos urbanos reciclables. El objetivo es crear rutas de recolección para cada día del mes, que se repetirán cada mes, minimizando el costo de operación. Dos de las características importantes del problema son la planificación de un período de tiempo relativamente largo y la recolección separada de tres tipos de residuos: (i) contenedores de vidrio, (ii) contenedores de papel, (iii) contenedores de plástico y metal. La operación de recolección se modeló de acuerdo con la práctica de la compañía que administra el sistema de recolección. Se desarrollaron técnicas heurísticas para resolver el modelo en tres fases: definición de las zonas geográficas atendido por los vehículos, definición del tipo de residuo a recolectar en cada día del mes, definición de las rutas de recolección. Los resultados preliminares sugieren que son posibles reducciones económicas significativas en los costos de recolección.

Inicialmente el WCVRPTW fue abordado por (Kim et al, 2006), se realizó un estudio sobre el problema de enrutamiento de vehículos para la recolección de basura comercial. El problema del enrutamiento de vehículos para la recolección de residuos sólidos con ventanas de tiempo se utiliza para encontrar rutas óptimas de recolección

para los camiones de basura de tal manera que todos los contenedores de basura se vacíen y los residuos se dirijan a sitios de eliminación respetando las ventanas de tiempo del cliente y asegurando que los conductores reciban los descansos que la ley requiere. Para alcanzar los objetivos, los autores desarrollaron un algoritmo constructivo basado en el algoritmo de Inserción de Solomon (Solomon, 1984), y para refinar la solución utilizaron un algoritmo clasificador basado en asociación (CBA) (Kim et al., 2006).

El WCVRPTW ha recibido cierta atención en los últimos años debido a que (Kim et al., 2006) abordaron un CVRPTW para la recolección de residuos sólidos en la vida real, en función de los viajes de eliminación múltiple y las pausas para el almuerzo de los conductores. Posteriormente, en 2007, (Ombuki et al., 2007) estudiaron el mismo problema WCVRPTW y propusieron un algoritmo genético multiobjetivo. En este trabajo, los autores utilizaron el conjunto de instancias propuesto por (Kim et al., 2006). (Benjamin & Beasley, 2010) propusieron un algoritmo constructivo goloso (greedy) y presentaron tres metaheurísticas: Tabu Search (TS), Variable Neighborhood Search (VNS), y Variable Neighbourhood Tabu Search (VNNTS), para la resolución del problema. Los autores utilizaron el mismo conjunto de instancias propuesto por (Kim et al., 2006). (Li, Borenstein & Mirchandani, 2008) consideran un problema de programación de enrutamiento de vehículos en el contexto de la recolección de residuos sólidos en la ciudad de Porto Alegre, Brasil. Se comprueba que el problema puede ser modelado como un caso especial del problema de enrutamiento de vehículos de un depósito único, que es poligonalmente solucionable. Sin embargo, debido a los beneficios sociales del sólido programa de manejo de residuos, es deseable obtener asignaciones equilibradas de viajes de recolección que descarguen su carga en las instalaciones de reciclaje. Un enfoque heurístico, que incorpora un el algoritmo de subasta y un método de penalización dinámica están diseñados para adquirir una buena solución. Finalmente, los experimentos computacionales fueron realizados en datos reales. Los resultados muestran que el enfoque heurístico reduce simultáneamente los costos totales y equilibra el número de viajes asignados a cada instalación de reciclaje. (Islam & Rahman, 2012) propusieron un algoritmo colonia de hormigas (ACO) para resolver el problema WCVRPTW. El conjunto de instancias utilizado fue propuesto por

(Kim et al., 2006) y los resultados obtenidos por el algoritmo ACO fueron mejor que los algoritmos propuestos por (Benjamin & Beasley, 2010). (Islam & Rahman, 2012) presentan un algoritmo basado en la colonia de hormigas para resolver este problema. Se realizaron extensos experimentos utilizando conjuntos de datos de instancias en la literatura y se comparó los resultados con las soluciones previas del mismo problema a partir de la literatura. Los resultados experimentales indican que el algoritmo proporciona mejores resultados que los algoritmos anteriores. Además, resulta que las metaheurísticas basadas en ACO pueden proporcionar este resultado dentro de un lapso de sólo media hora. Se observó que el algoritmo sería muy adecuado para este problema (Islam & Rahman, 2012).

Buhrkal et al. (2012) presentaron un modelo de programación lineal completa (MILP) del problema WCVRPTW, y propusieron un algoritmo de búsqueda de vecindades grandes (ALNS) para resolver el problema. Se ha probado el algoritmo en el conjunto instancias propuesto por Kim et al. (2006) y en un conjunto de instancias proporcionadas por una empresa de recolección de basura danesa. Los experimentos realizados por el algoritmo ALNS fueron mejores que los propuestos por (Benjamin & Beasley, 2010) e (Islam y Rahman, 2012). (Benjamin & Beasley, 2013) mejoraron la efectividad de los algoritmos, produjeron soluciones de mejor calidad al minimizar la distancia de viaje utilizando una búsqueda tabú y una búsqueda de vecindad variable, y una combinación de estas.

(Das & Bhattacharyya, 2015) En este artículo, se enfoca a la recolección y gestión de los desechos sólidos. Esta problemática se origina principalmente con el fin gestionar eficazmente los residuos. La problemática abordada principalmente considera en que las fuentes de generación de los residuos se presentan de forma dispersa al realizar la gestión de los residuos para minimizar el tiempo y costo de traslado. La investigación se enfoca principalmente en la administración de residuos en tres etapas. Cada etapa ha sido optimizada con la ayuda de TSP. Propone una solución heurística para una óptima recolección de residuos y problema de transporte. La propuesta calcula la recogida y transporte óptimo de residuos. Se ha demostrado la eficacia del esquema

propuesto. Los resultados demuestran que el esquema propuesto es capaz de reducir más del 30% de la longitud total de la ruta de recogida de residuos. Esta cuanto a la minimización de la distancia de recorrido total del camino. En la recolección y transporte de residuos se determina un ahorro monetario consistente en las operaciones de gestión de residuos.

En (Markov, Varone & Bierlaire, 2016) se propone un modelo general de enrutamiento de vehículos que incluye múltiples estaciones de disposición, múltiples depósitos y vehículos no homogéneos y ventanas de tiempo para la colección, se tiene en cuenta viajar en rutas de ida, el número de vehículos por m^2 y el tiempo de espera en las paradas de tráfico para minimizar el tiempo de operación. El modelo propuesto podría usarse para escenarios que tienen estructuras de nodos y características de vehículos similares. Un estudio de caso en la ciudad de Danang, Vietnam, se da para ilustrar la aplicabilidad de este modelo. Los resultados experimentales muestran que el nuevo modelo reduce tanto las distancias totales de viaje como las horas operativas de los vehículos en comparación con las de los escenarios prácticos. Rutas óptimas de vehículos en calles y mercados en Danang. Esos resultados son significativos para los profesionales y los responsables políticos locales.

(Idrus et al, 2017) presentaron una revisión bibliográfica de las principales contribuciones del Problema de Enrutamiento de Vehículos para recolección de residuos en los últimos 11 años. Los objetivos fueron identificar la función objetivo del problema, las restricciones, las heurísticas y el conjunto de instancias utilizado en cada uno de los problemas presentados en los estudios de caso.

(Rabbani, Farrokhi-Asl & Asgarian, 2017) desarrollan un diseño separado de la ubicación de los depósitos y el enrutamiento para el servicio a los clientes. El problema de la recolección de desechos se considera con respecto a las funciones objetivas económicas y sociales. Se utiliza un algoritmo genético de clasificación no dominado (NSGAI) para ubicar depósitos e instalaciones de tratamiento y se diseñan las rutas partiendo de los depósitos para atender a los clientes. Se propone un nuevo modelo

matemático y dos funciones objetivos incluyendo objetivo económico (costo de apertura de depósitos y facilidad de tratamiento y costo de transporte) y objetivo social; es decir, impacto negativo del tratamiento. Se aplica una representación de solución basada en orden directo para codificar soluciones del problema y un enfoque de agrupamiento se utiliza para generar soluciones iniciales adecuadas.

(Hannan, Akhtar, Begum, Basri, Hussain & Scavino, 2018) aplicaron un modelo de problema de enrutamiento de vehículos con capacidad limitada para la calendarización de recolección de residuos sólidos y optimización de rutas utilizando el algoritmo PSO. Se utilizan el nivel de pérdida de umbral (TWL) y los conceptos de programación se aplican en el modelo CVRP basado en PSO bajo diferentes conjuntos de datos. Los resultados obtenidos de diferentes conjuntos de datos muestran que el modelo algorítmico CVRP propuesto proporciona la mejor recolección de desechos y optimización de rutas en términos de distancia de viaje, desperdicio total, eficiencia de recolección de desechos y estanqueidad al 70–75% de TWL. Los resultados obtenidos para una programación de la semana muestran que el 70% de TWL se desempeña mejor que todas las consideraciones de nodos en términos de desechos recolectados, distancia, estrechez, eficiencia, consumo de combustible y costo. El modelo optimizado propuesto puede servir como una herramienta valiosa para la recolección de desechos y la optimización de rutas para reducir los impactos socioeconómicos y ambientales.

Tabla 2.3. Trabajo relacionado al ruteo vehicular

Año	Autores	Algoritmos	Metodología
2000	Tung and Pinnoi	Inserción heurística de Solomon	Modificó el algoritmo de inserción de Solomon y lo aplicó a una recolección de residuos sólidos en Hanói, Vietnam.
2004	Teixeira	Heurística para PVRP	PVRP para la recolección por separado de tres tipos de residuos: vidrio, papel y plástico / metal.

2006	Kim	Inserción heurística de Solomon	Recolección de instancias reales de desechos con enfoque VRPTW, teniendo en cuenta los viajes de múltiples sitios de recolección y los tiempos para el almuerzo de los conductores.
2006	Nuortio	Metaheurística de vecindad variable umbral	Optimización de las rutas de los vehículos y los horarios para la recolección de residuos sólidos municipales en el este de Finlandia.
2007	Ombuki	Algoritmo genético multi-objetivo	Algoritmo genético multi-objetivo en un conjunto de datos de referencia de problemas del mundo real obtenidos por (Kim et al., 2016)
2008	Li	Heurística	Diseño de horarios de camiones diarios, recolectan desechos sólidos en rutas fijas y cargas vacías en una de varias operaciones instalaciones de reciclaje en el sistema en la ciudad de Porto Alegre, Brasil.
2010	Benjamin and Beasley	Búsqueda tabú/VNS	Mejóro los resultados al minimizar la distancia de viaje utilizando una búsqueda tabú y una búsqueda de vecindad variable y una combinación de estos.
2012	Katja Buhrkal	ALNS	El algoritmo ALNS mejoró los resultados del conjunto de instancias de la literatura de (Kim et al., 2016), así como para los casos proporcionados por una empresa danesa de recolección de basura.
2012	Islam & Rahman	ACO	El algoritmo de ACO mejoro los resultados que obtenidos por los algoritmos propuestos por (Benjamin & Beasley, 2010). Este estudio está enfocado a la recolección de residuos sólidos en la vida real.
2015	Das & Bhattacha	SA	Problema del transporte de residuos para

			diseñar sistemas de gestión de residuos rentables y de reducción de tiempo. Se dividió todo el sistema de gestión de residuos en tres etapas. Cada etapa ha sido optimizada con la ayuda de TSP, con una heurística.
2017	Idrus et al.	Revisión	Revisión bibliográfica de las principales contribuciones del Problema de Enrutamiento de Vehículos para recolección de residuos en los últimos 11 años
2018	Hannan et al.	PSO	Proporciona mejor recolección de desechos y optimización de rutas en términos de distancia de viaje, desperdicio total, eficiencia de recolección de desechos y estanqueidad al 70–75% de TWL
2020	Tesis	SA/TS/LS	Proporciona mejor recolección de desechos y optimización de rutas en términos de distancia de viaje, desperdicio total, eficiencia de recolección de desechos y estanqueidad al 70–75% de TWL

En la **tabla 1.4** se puede apreciar el resumen de la evolución de la adaptación del problema de enrutamiento de vehículos enfocado a la recolección de residuos.

2.15.1. Algoritmos utilizados para resolver el WCVRPTW

En esta sección, mostramos un resumen de los diferentes algoritmos aproximados utilizados para resolver diferentes tipos de WCVRPTW. Como primer paso, estamos considerando el uso de recocido simulado para abordar el problema de la recolección de basura, luego implementar más algoritmos para encontrar mejores soluciones, hacer comparaciones con los resultados dentro de la literatura, elegir el algoritmo que obtuvo los mejores resultados para adaptarlo para resolver la problemática.

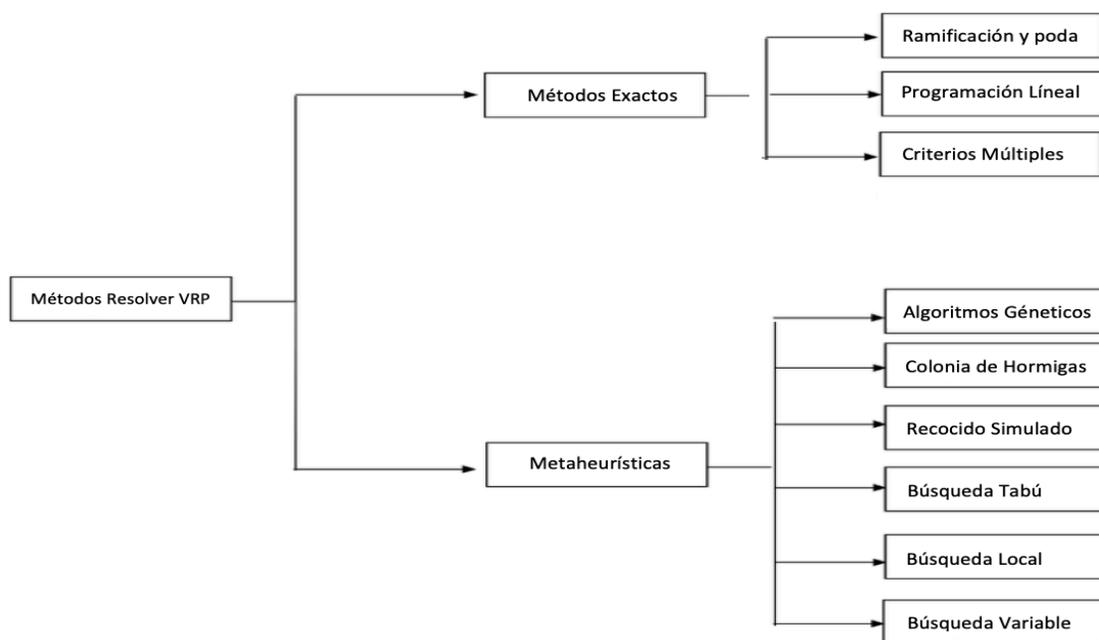


Figura 2.22. Algoritmos para resolver el WCVRPTW (Fuente Propia)

El resumen de las mejores heurísticas con los mejores resultados reportados en la literatura para resolver el WCVRPTW se ilustra en la *figura 2.22*.

Tabla 2.4. Procedimientos de solución propuestos para el WCVRPTW.

Tipo de enfoque	Procedimiento de solución
Solución exacta	Formulaciones integrales de programación lineal Ramificación y corte
Heurísticas	Heurística simple Algoritmos evolutivos Inserción de Solomon ACO Recocido simulado Búsqueda de tabú Algoritmos genéticos
Transformaciones	Múltiples centros de disposición a uno solo

En la **Tabla 2.4** se muestran las variaciones que se han encontrado para resolver la problemática de la recolección de residuos sólidos.

Capítulo 3. Metodología para la resolución del problema

Este capítulo proporciona los antecedentes relevantes sobre VRPTW, sus variantes para resolverlo y su formulación matemática. Finalmente, se ofrece una visión general del algoritmo, la estructura de las fases del algoritmo y la optimización de las rutas. Este capítulo proporciona los detalles de implementación del algoritmo propuesto para resolver las instancias del VRPTW. El objetivo de este capítulo es generar un modelo matemático que contenga la abstracción del problema del VRPTW, eliminando las complejidades y haciendo suposiciones pertinentes; aplicando una técnica matemática y haciendo una representación simbólica de la realidad del problema que se va a solucionar. Forma expresiones lógicas matemáticas, con la finalidad de obtener el costo mínimo en la función objetivo del modelo.

3.1. Planteamiento

En esta sección definimos el problema en estudio y la notación utilizada en todo el capítulo para el desarrollo del modelo matemático, primeramente, se han identificado los requisitos y especificaciones con las que cuentan las empresas del sector de servicios de recolección de residuos sólidos en el estado de Cuernavaca. Dentro de estos aspectos se han considerado la capacidad de los vehículos, el número de clientes, el tiempo de servicio de cada cliente, la capacidad máxima de recolección permitida y entre otros elementos que permiten la formulación del modelo. Las decisiones del modelo están relacionadas con la asignación de rutas de recolección para la demanda por un conjunto de clientes. El objetivo del modelo es la minimización de los costos de transporte (suma de los costos asociados a la distancia de recorridos del recorrido de la recolección de residuos sólidos de los clientes en el sector turístico y el tiempo de servicio).

3.1.1. Definición del modelo matemático

Formalmente el problema se define como el problema de enrutamiento de vehículos de recolección de residuos (WCVRPTW). El problema del WCVRPTW (Kim BI, Kim, Sahoo, 2006), difiere del VRPTW tradicional en que los vehículos recolectores de basura deben vaciar su carga en los sitios de disposición. Los vehículos deben estar vacíos cuando vuelvan al depósito. Se permiten viajes múltiples a los sitios de disposición para los vehículos. Pero para el problema específico de la recolección de residuos sólidos en el sector turístico en Cuernavaca el problema se resuelve como VRPTW debido a que el centro de disposición y el depósito es el mismo lugar, es decir que en el depósito se trata los residuos y se almacenan los vehículos.

El problema es representado por un grafo completo no dirigido ponderado $G = (V, E)$, con un conjunto de vértices $V = \{V_d, 1, 2, \dots, V_n, V_d\}$ y un conjunto de aristas E . Consiste en un conjunto de clientes V_c , con todos los pesos positivos; El nodo $V_d \in V$ representan el depósito, es decir, del cual todos k vehículos parten los recorridos pasando por cada vértice exactamente una vez, y finalmente regresando al depósito, a esto se le llama tour. La longitud de un tour es la suma de las longitudes de las aristas en el camino que define el recorrido. Ningún arco termina en el nodo V_d del depósito y todas las rutas comienzan y terminan en el depósito V_d . El conjunto de vértices representado por los clientes $V_d \in V$ donde $V \setminus V_d$ y $V_c = \{V_1, \dots, V_n\}$; y los demás vértices son clientes V_c ; El conjunto de n vértices que denotan a los clientes se denota como N .

El conjunto de arcos A denota todas las conexiones posibles entre los nodos (incluido el nodo que denota el depósito). E representa el conjunto de las parejas (i, j) : donde i, j pertenece $\in V$ al conjunto de los vértices se le denomina el conjunto de aristas. $E = \{(V_i, V_j) : i \neq j\}$ es el conjunto de arcos, cada una con un costo de viaje no negativo C_{ij} . Asociamos un costo C_{ij} asociado a la distancia o tiempo de viaje definido para cada arista del conjunto de aristas y un tiempo t_{ij} con cada arco $(i, j) \in A$ de la red de enrutamiento. El tiempo de viaje t_{ij} , j puede incluir el tiempo de servicio en el cliente i . Cada cliente $V_c \in V \setminus \{V_d\}$ tiene una demanda no negativa d_i , $i \in C$; Cada ruta

tiene un límite diario de carga máxima de recolección de basura permitida R por cada cliente V_c a la cual debe ser satisfecha por un vehículo y la capacidad C del vehículo debe respetarse; Una flota homogénea con cada vehículo con capacidad Q , es disponible en el depósito V_d .

t_{ij} y C_{ij} representan el tiempo de viaje y el costo asociado con el arco (i, j) . Respectivamente cada nodo $i \in V_c$ tiene un tiempo de servicio asociado s_i y una ventana de tiempo $[a_i, b_i]$ y definimos el arco d_i como la cantidad de recolección de un cliente $i \in V_c$. Un vehículo puede llegar antes del comienzo de la ventana de tiempo (es decir, a_i), lo que significa que incurre en tiempo de espera hasta que el servicio sea posible. Sin embargo, ningún vehículo puede llegar más allá del cierre de un intervalo de tiempo dado b_i . Los vehículos también deben salir del depósito dentro de la ventana de tiempo del depósito $[a_0, b_0]$ y deben regresar antes o en el momento $b_n + 1$. Suponiendo que el tiempo de espera se permite sin costo, podemos suponer que $a_0 = b_0 = 0$; es decir, todas las rutas comienzan en el tiempo 0.

Para modelar el problema del depósito, se divide en un depósito inicial y final $\{0,0\}$. El problema puede luego ser modelado usando tres tipos de variables: $X_{ijk} \in \{0,1\}$, $A_{l,k} \in \{0,1\}$, $w_{ik} \in \mathbb{R}$, $i \in V$, $k \in K$, $l \in K$, X_{ijk} es uno si y sólo si el vehículo l de K usa el arco $(i, j) \in E$, d_i representa la demanda acumulada en el nodo $i \in V$ para el vehículo $l \in K$, y w_i representa la hora de inicio del servicio en el nodo $i \in V$ para el vehículo $l \in K$. $w_{ik}, i \in V, k \in K$, especificando el inicio del servicio en el nodo i cuando es atendido por un vehículo k (Kim, Kim, Sahoo, 2006).

En cuanto a las consideraciones; Toda ruta comienza y termina en el depósito; Todo vehículo debe salir y volver vacío al depósito; La capacidad del vehículo debe respetarse; Cada cliente es atendido solamente una vez por solamente un vehículo; Todos los clientes deben ser atendidos dentro de su respectiva ventana de tiempo; Cada conductor del vehículo debe realizar una sola pausa para el almuerzo dentro de la ventana de tiempo; Cada ruta tiene un límite diario de paradas (atención al cliente); Cada ruta tiene un límite diario de carga (volumen de basura recogido).

La meta del WCVRPTW es generar un número de recorridos de los vehículos entre el depósito y los clientes para satisfacer la recolección de la demanda de los clientes tal que la distancia total sea la mínima cumpliendo con las restricciones para construir un problema WCVRPTW (Kim et al., 2006). Para el VRPTW las restricciones básicas del problema consisten en (Desrosiers, Dumas, M. Solomon, 1995) cada sitio es visitado una sola vez por un solo vehículo; cada ruta se inicia en el mismo depósito; todas las demandas de todos los clientes deben ser satisfechas. El vehículo no debe llegar antes de tiempo y no más tarde de tiempo. Un vehículo que llega antes de tiempo producirá un tiempo de espera. Un vehículo que llegue después de la hora incurrirá en un retraso. Todos los vehículos programados tienen que regresar al depósito dentro del tiempo máximo de operación para cada vehículo.

3.1.2. Notación Matemática

Una notación matemática es un sistema de escritura utilizado para registrar conceptos en matemáticas. La notación utiliza símbolos o expresiones simbólicas que pretenden tener un significado semántico preciso.

Tabla 3.1. Notación de objetos

Notación	Representación	Definición
V	$\{V_d\} \cup V_c$	Vértices del grafo $V = \{V_d \{v_0\}, \{V_1, \dots, V_n\}$ $ V =n+2$
V_d	$\{V_0\}$	Conjunto de depósito representado indexados por i, j
V_c	$\{V_1, \dots, V_n\}$	Conjunto de vértices considerando los puntos de servicio
A	$\{(i, j): i, j \in V\}$	Conjunto de aristas. Donde el costo asociado a la arista (i, j) es C $(i, j) \in E, i, j \in V$ y $ E = n^2 - n$.
K	$\{1, \dots, k\}$	Conjunto K de vehículos

N	$V_i \in V, i \in \text{Enteros y}$ $1 \leq i \leq n$	Representa el número de vértices
-----	--	----------------------------------

En la **Tabla 3.1** se muestra la notación que el modelo matemático considera (Kim et al., 2006). A continuación, cada uno de estos objetos básicos se presenta brevemente.

Mapa. Este objeto, es el plano utilizado para el transporte de bienes, generalmente se describe como un grafo ponderado. Los vértices representan la posición, es decir uniones de las calles, el depósito y los clientes, mientras que los arcos representan conexiones entre ubicaciones. Cada arco está asociado con un costo, y una venta de tiempo.

Clientes. Son las entidades que requieren servicio. Cada cliente cuenta con una demanda de servicio, cantidad de bienes o materiales a ser entregado o recogido. El problema es dado por un conjunto de clientes. $V_c = \{1, 2, \dots, n\}$, residiendo en n diferentes ubicaciones. Cada par de locaciones (i, j) , donde $i, j \in V_c$ y $i \neq j$, es asociado con un tiempo de viaje t_{ij} y una distancia de recorrido c_{ij} que son simétricas ($t_{ij} = t_{ji}$ y $d_{ij} = d_{ji}$). Denotamos por d_i , la demanda en el cliente i . Tiempo de servicio (carga / descarga) s_i , Hora de inicio del servicio a_i deseada y hora de fin del servicio b_i (con penalizaciones)

Los clientes tienen varias propiedades:

- El vértice en el grafo de las calles donde se encuentra el cliente;
- Cantidad de demanda que debe ser entregada o recolectada en el cliente;
- Intervalo durante el cual se debe atender al cliente (ventana de tiempo);
- Tiempo requerido para servir al cliente (el tiempo de carga o la descarga);
- Subconjunto de vehículos que puede servir al cliente.

Depósito. El depósito central se denota por 0.

Los vehículos. Los clientes son atendidos desde un depósito con una flota homogénea y limitada. Los vehículos salen y vuelven al depósito. Hay un conjunto K de vehículos, $K = \{1, \dots, k\}$, con capacidades idénticas. La capacidad de cada vehículo $k \in K$ es representado por d_{ik} . Sea $X_{ijk} = \{V_1, \dots, V_n\}$ la ruta para el vehículo i , donde X_{ijk} es el índice del cliente j que visitó y u_i es cantidad de clientes en la ruta. Asumimos que cada ruta termina en el depósito, $X_{0jk} = 1$. Capacidad del vehículo (carga máxima permitida; peso, volumen). Costos (asociados con kilometraje, tiempo, combustible, viaje, carga).

Ventana de Tiempo. Cada cliente $i \in V_c$ tiene una ventana de tiempo, es decir un intervalo en el cual puede ser atendido $[a_i, b_i]$, donde $a_i \leq b_i$, cual corresponde, respectivamente, al primer y último momento en que se inicia el servicio al cliente i . Sea s_i el tiempo de servicio al cliente i . Una ventana de tiempo, lo que significa que cada cliente i tiene un tiempo predefinido con una hora de llegada a_i (límite inferior) y una hora de cierre b_i (límite superior) para la visita de vehículos. Los vehículos deben llegar antes del límite superior de la ventana b_i y si llegan antes del límite inferior a_i deben tener un tiempo de espera w_i (tiempo de espera). Cada cliente impone un tiempo de servicio s_i que corresponde al tiempo de carga / descarga de la recolección. La ventana de tiempo del depósito significa que cada vehículo que sale del depósito a la vez a_0 vuelve al depósito antes del tiempo b_0 .

Este objeto tiene varias propiedades:

- El vértice en el grafo del depósito es donde parten;
- Capacidad del vehículo, carga máxima del vehículo.
- Subconjunto de arcos del grafo de calles que puede atravesar el vehículo; el costo asociado al uso del vehículo (por ejemplo, por unidad de tiempo o distancia).

Ruta. Forma de visitar todas las ciudades o clientes, es simplemente el orden en que se visitan las ciudades; el orden se llama una tour o circuito por las ciudades.

Costo. El costo de atravesar un arco (i, j) , c_{ij} , se define como el producto de la distancia del arco (i, j) y flujo en este arco.

Restricciones locales (ruta única)

- Capacidad del vehículo
- Distancia / duración máxima permitida de la ruta
- Limitaciones de tiempo (llegada, salida, ventanas de tiempo)
- Tipo de servicio (recogida, entrega o ambos)
- Prioridad entre los clientes:
- Recogida y entrega

Restricciones globales (conjunto completo de rutas)

- Número máximo de vehículos
- Número máximo de rutas (para vehículo o depósito)
- equilibrio de la carga de trabajo del vehículo
- Periodos de trabajo y turnos (tiempo mínimo entre rutas).

3.1.3. Parámetros de los objetos

Los parámetros de los objetos son variables locales disponibles dentro de todas las funciones. Puede hacer referencia a los argumentos de una función dentro de esa función utilizando los atributos del objeto. Los parámetros representan los valores conocidos del problema. Para resolver el problema de VRPTW se definieron los atributos para cada objeto.

Tabla 3.2. Tabla notación del modelo matemático

Notación	Definición
i, j	Índices de nodo, $i, j = 1, \dots, n$
d_i	Demanda del cliente i (nodo),

c	Capacidad de los vehículos (peso máximo),
t_{ij}	Tiempo de viaje entre el cliente i y j
C_{ij}	Costo asociado de viaje entre el cliente i y j
s_i	Tiempo de servicio del cliente i ,
x_i	Coordenada en X (Longitud) del cliente i
y_i	Latitud del cliente i
$[a_i, b_i]$	Ventana de tiempo para atender al cliente i ,
R	cantidad máxima de basura que se puede recoger,
A	Conjunto formado por los pares ordenados i, j
a	Tiempo de inicio de la ventana en un nodo
b	Tiempo límite de la ventana en un nodo
C	Costo de servicio total
K	flota de vehículos
k	Vehículo

En la **Tabla 3.2** los parámetros de los objetos son basados en la definición de (Kim et al.,2006).

3.1.4. Variables de decisión

Las variables de decisión usualmente son determinadas a partir de la solución de un problema o modelo.

Tabla 3.3. Tablas variables de decisión del modelo matemático

Variable	Valor	Significado
X_{ijk}	1	si la arista (i, j) es usado por el vehículo k
	0	De lo contrario
Y_{ijk}	1	si la arista (i, j) es visitado por el mismo vehículo k
	0	De lo contrario
F_{ijk}	1	si la arista i es el primer nodo visitado por el tour del vehículo
	0	De lo contrario
Z_j	1	si la arista i es el último nodo visitado por el tour del vehículo
	0	De lo contrario
$W_{i,k}$		tiempo que el vehículo k inició el servicio en cliente i ; $\forall i \in V; k \in K$

$d_{i,k}$	demanda acumulada de cliente y no vehículo k ;
u_i	Número de nodos visitados en cada ruta hasta el nodo $i \quad \forall i \in V$

Las variables de decisión que estamos considerando para nuestro problema se representan en la **Tabla 3.3** (Kim et al., 2006):

3.1.5. Restricciones

Las restricciones del modelo es correspondencia de los recursos disponibles y entre las variables de decisión. El modelo cuenta con restricciones que reducen el valor de las variables de decisión. Se imponen estas restricciones para garantizar la factibilidad de las soluciones. En el modelo se incluye, adicionalmente de las restricciones, la restricción de no negatividad de las variables de decisión.

Las siguientes restricciones son impuestas (Kim et al., 2006):

- El vehículo inicia la ruta desde el depósito y vuelve al depósito.
- Costo asociado de viaje entre el cliente i y j (i, j) tiene que ser mayor a 0
- Cada cliente debe ser visitado por un solo vehículo. (Cada cliente $V_c \in V \setminus \{V_0\}$ es visitado exactamente por un vehículo;)
- Restricciones de ventana de tiempo.
- El vehículo debe partir y volver vacío al depósito.
- La demanda acumulada de recolección de todos los clientes no debe exceder la capacidad del vehículo.

3.1.6. Modelo Matemático

Para resolver el problema podemos establecer el VRPTW en la siguiente forma matemática como Solomon había usado un MILP (Hillier, 2012) es un método utilizado para obtener un resultado óptimo en un modelo matemático en el que todas las funciones matemáticas son lineales. El modelo matemático consiste en una función

lineal para optimizar (minimizar o maximizar), restricciones lineales y, por lo general, variables no negativas. Los parámetros y variables utilizados en el modelo MILP introducido se presentan aquí seguidos del modelo matemático y el método de preprocesamiento.

El objetivo del modelo es minimizar la distancia total recorrida respetando las restricciones de ventana de tiempo.

$$\min \sum_{(i,j) \in A}^n C_{ij} \sum_{k \in K}^n X_{ijk} \quad (3.1)$$

S.A

$$\sum_{i \in V_c}^n \sum_{k \in K}^n X_{ijk} = 1 \quad \forall j \in V_c \quad (3.2)$$

$$\sum_{j \in V}^n X_{0jk} = 1 \quad \forall k \in K \quad (3.3)$$

$$\sum_{i \in V}^n X_{i0'k} = 1 \quad \forall k \in K \quad (3.4)$$

$$\sum_{i \in V_c}^n x_{ijk} = \sum_{i \in V_c}^n x_{jik} \quad \forall j \in V_c, k \in K \quad (3.5)$$

$$\sum_{i \in V}^n x_{ik}, n + 1, k = 1 \quad \forall k \in K \quad (3.6)$$

$$w_{ik} + s_i + t_{ij} - w_{jk} \leq (1 - X_{ijk}) M \quad \forall (i, j) \in A, k \in K \quad (3.7)$$

$$E \leq w_{ik} \leq L \quad \forall k \in K, i \in (0, n + 1) \quad (3.8)$$

$$a_i \sum_{i \in V_c}^n X_{ijk} \leq W_{ik} \leq b_i \sum_{i \in V_c}^n X_{ijk} \quad \forall i \in V_c, k \in K \quad (3.9)$$

$$\sum_{i \in V_c}^n d_i \sum_{k \in K}^n d_{ijk} \leq C \quad \forall k \in K \quad (3.10)$$

$$d_{ik} + d_i \leq C + (1 - x_{ijk})M \quad \forall i \in V_c \quad (3.11)$$

$$\sum_{i \in V_c}^n d_i \leq R \quad \forall i \in V_c \quad (3.12)$$

$$X_{ijk} \geq 0 \quad \forall i \in V_c, k \in K \quad (3.13)$$

$$X_{ijk} \in \{0, 1\} \quad \forall (i, j) \in A, k \in K \quad (3.14)$$

La ecuación (3.1) minimiza el costo total de viaje a seguir por cada vehículo k . El costo total se puede interpretar como el recorrido total en unidades distancia o tiempo.

Las restricciones del modelo son las siguientes:

La ecuación (3.2) indica que el recorrido x del nodo i a j que forma la ruta debe ser igual a 1 para garantizar un vehículo por cada ruta. La ecuación (3.3) proporciona el conjunto de restricciones de flujo que aseguran que el número de clientes j son directamente conectados desde el depósito de origen al iniciar el recorrido utilizando el vehículo k , sea igual a uno. Partiendo del depósito solo un nodo destino puede ser atendido. El vehículo inicia la ruta desde el depósito. El vehículo vuelve al depósito al final de la ruta. La ecuación (3.4) proporciona el conjunto de restricciones que aseguran que el vehículo debe de volver al depósito al finalizar la ruta. La ecuación (3.5) proporciona las restricciones que el flujo de entrada y salida deben ser iguales, excepto los nodos de depósito. La ecuación (3.6) proporciona el conjunto de restricciones que garantiza que para cada vehículo k solo debe existir un nodo que conecte directamente con el depósito $n+1$. La ecuación (3.7) proporciona el conjunto de restricciones que garantiza la atención del nodo i antes del nodo j . Considerando que la suma del tiempo de inicio de servicio al nodo i , el tiempo de servicio asignado al nodo i , el tiempo de recorrido del nodo i al nodo j y el tiempo de inicio de servicio del nodo j , debe ser menor o igual a $(1 - X_{i0k})$. Para garantizar dichas desigualdades se asigna una penalización tan grande, representada por M en la ecuación (8), para obligar a la factibilidad de la solución, de tal forma que si X_{i0k} toma valor de 0 la desigualdad siga cumpliéndose. La restricción (3.8) garantiza que se respete el conjunto de restricciones que indiquen el tiempo de inicio de servicio W_{0k} en el depósito no debe exceder los límites de la ventana de tiempo del nodo cero definidos por el intervalo $[E, L]$. La restricción (3.9) garantiza que se respete el conjunto de restricciones que indiquen el tiempo de inicio de servicio W_{ik} al nodo i no debe exceder los límites de la ventana de tiempo del nodo i definido por el intervalo $[a_i, b_i]$. La restricción (3.10) garantiza que el conjunto de restricciones representadas en las desigualdades es la suma de las demandas d_i de cada nodo en el recorrido no debe exceder la capacidad C total

del vehículo k . Donde N indica el número de clientes a atender, k indica la flota de vehículos con capacidad C . La ecuación (3.11) proporciona el conjunto de restricciones que garantiza que la atención del nodo i antes del nodo j . Considerando que la suma de la demanda del vehículo k , la demanda asignada al nodo i , debe ser menor o igual a la capacidad del vehículo ($C - X_{i0k}$). Para garantizar dichas desigualdades se asigna una penalización tan grande, representada por M en la ecuación (8), para obligar a la factibilidad de la solución, de tal forma que si X_{i0k} toma el valor de 0 la desigualdad siga cumpliéndose. La restricción (3.12) garantiza que el conjunto de restricciones representadas en las desigualdades que la demanda d_i del cliente i no exceda la cantidad máxima de basura que se puede recoger. La restricción (3.13) garantiza que el modelo lineal se defina como un modelo lineal entero de variables binarias. La restricción (3.14) garantiza que el modelo lineal se defina como un modelo lineal de no negatividad tomando un modelo lineal entero de variables binarias.

3.2. Descripción del algoritmo para resolver el VRPTW

En esta tesis, un algoritmo de recocido simulado se emplea para abordar el problema del VRPTW para minimizar la distancia total recorrida utilizando operadores que combinan variaciones de conceptos clave inspirados en técnicas de enrutamiento y estrategias de búsqueda utilizadas para una variante temporal del problema para proporcionar una búsqueda adicional mientras se equilibra explotación y diversificación. Las técnicas de búsqueda de algoritmos híbridos representan algunas de las metaheurísticas más eficientes para abordar el VRP y / o sus variantes. La idea básica es permitir la selección de soluciones peores una vez que se ha obtenido un óptimo local alcanzado.

3.2.1. Representación de la solución

Hay diferentes representaciones disponibles para la solución y la selección de la representación adecuada es específica del problema. La buena representación es lo que hace que el espacio de búsqueda sea más pequeño y, por lo tanto, una búsqueda más

fácil. Nosotros optamos por utilizar la representación de permutación, es la más común para problemas de ruteo como el problema del agente viajero: $X_0 = [x_1, x_2, x_3, \dots, x_n]$.

La inicialización de la solución consiste en colocar a cada uno de los vehículos del conjunto de clientes a visitar, esta asignación de forma aleatoria o mediante algún algoritmo voraz que nos permita generar la solución inicial. Para cada k vehículo su posición i se denota a través de un vector $X_{ijk} = [x_1, x_2, x_3, \dots, x_n]$ donde j representa de 1 hasta n clientes, que es el tamaño del vector el cual será utilizado para expresar la solución. La codificación de la solución consiste en cadenas de 1 a n número de clientes que se generan de forma aleatoria a través de la variable X_{ijk} , esta variable toma un valor de un número real dentro del intervalo de $[1, n]$. La representación de ruta que se utiliza para codificar la solución de VRPTW se muestra en la **figura 3.1**.

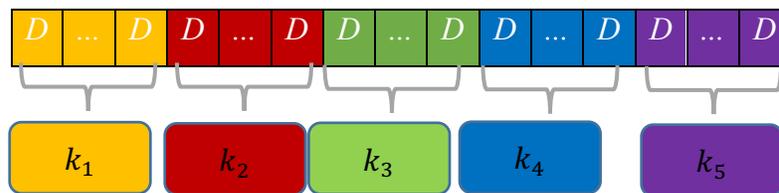


Figura 3.1. Representación de la asignación de clientes a los vehículos

3.2.2. Solución Inicial

Para resolver efectivamente el problema VRPTW, es necesario para obtener una solución inicial factible en la que todas las restricciones sean satisfechas. Debido al hecho de que encontrar de una solución factible para el problema VRPTW con el mínimo número de vehículos es una tarea muy compleja, los algoritmos heurísticos constructivos a menudo producen soluciones de mala calidad que solo sirven como punto de partida para una mayor optimización. La solución inicial generalmente se crea con algunas heurísticas de inserción más baratas. Se construye una solución inicial al intentar utilizar completamente un vehículo durante el día en su jornada laboral (con el objetivo de minimizar el número total de vehículos utilizados). Una vez que un vehículo

no se puede usar más, entonces comenzamos una nueva ruta de vehículo con un vehículo nuevo. Para la asignación se agregan los nodos con su ventana de tiempo más próxima, lo antes posible de acuerdo con su período de tiempo. Sin embargo, nosotros estamos adaptando un algoritmo voraz clásico a un algoritmo aleatorio, es decir que cada vez que genera una asignación de clientes para cada vehículo esta asignación se realiza de forma aleatoria con forme a una métrica, es decir ordenar los clientes bajo un criterio. El primer criterio de ordenamiento es ordenar los clientes más próximos atender por su ventana de tiempo a_i , el segundo criterio es ordenar los clientes que su ventana de tiempo sea la más próxima a cerrarse b_i y por último ordenar los clientes con mayor demanda. Esto nos permite tener variaciones de la solución inicial para tratar de mejorar la solución inicial. En la **figura 3.2** se puede apreciar una ilustración de cómo se realiza a selección.

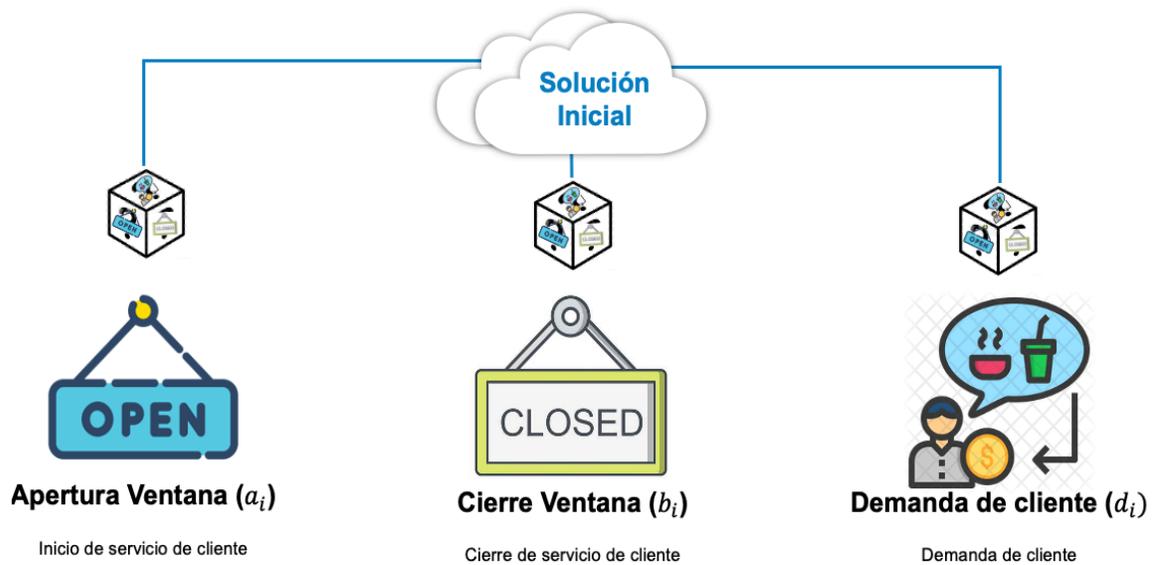


Figura 3.2. Criterios de asignación de clientes a vehículos.

Una vez que se seleccionó el criterio de ordenamiento de los clientes, se procede asignar los clientes al vehículo, una vez que el vehículo no pueda atender al cliente por cuestiones de las restricciones, se pasa al siguiente vehículo y se vuelve a generar el criterio de ordenamiento, este procedimiento se repite hasta que todos los clientes sean visitados. En la **figura 3.3** se puede apreciar la asignación de los clientes al k vehículo.

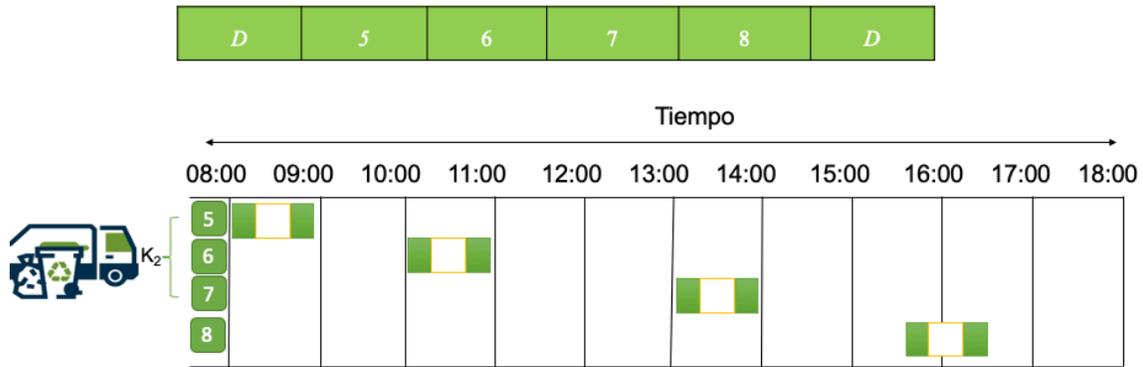


Figura 3.3. Asignación de los clientes al k vehículo

A continuación se presenta nuestro pseudocódigo para realizar la asignación de las rutas al vehículo, cada vez que genera una asignación de clientes para cada vehículo esta asignación se realiza de forma aleatoria con forme a una métrica, por apertura de ventana de tiempo, cierre de ventana de tiempo o por demanda es decir ordenar los clientes bajo un criterio Si los clientes están representados por las ventanas de tiempo, son primero ordenados en orden creciente de las ventanas más próximas hacen con el depósito. A continuación, la asignación de las m rutas son entonces construidas con el siguiente procedimiento:

1. Elegir aleatoriamente un criterio de asignación $\{a_i, b_i, d_i\}$
2. Ordenar el conjunto de clientes $\{1, \dots, n\}$ bajo el criterio seleccionado
3. Elegir el primer cliente $j \in \{1, \dots, n\}$
4. Conjunto $k := 1$
5. Usando la secuencia de clientes $j, j + 1, \dots, n, 1, \dots, j - 1$, realice los siguientes pasos para cada cliente i
 - Si la inserción del cliente i en la ruta k resultaría en la violación de las restricciones de carga o duración, conjunto $k := \min \{k + 1, m\}$
 - Inserte el cliente i en la ruta k para minimizar el aumento en el tiempo total de viaje de la ruta k
6. Repetir hasta que todo el conjunto V_c de clientes sean atendidos

Este proceso de asignación de rutas se repite hasta que todos los clientes sean atendidos. El cliente i es el cliente que tiene el menor tiempo de viaje del cliente r en el

final actual de la ruta, siempre que cumpla con las restricciones de ventanas de tiempo y de demanda mencionadas anteriormente. Esta expresión es relativamente compleja. Aquí consideramos solo aquellos clientes j de tal manera que cuando el vehículo llegue a j (en el momento $T + E_0$) será posible atender al cliente ya que la visita caerá en su ventana de tiempo $[E_j, E_j]$ y la carga que se recogerá en j se ajustará al vehículo (en términos de la ruta actual, el trabajo de todo el día y el número total de clientes visitados). Además, j tiene que ser un cliente de modo que si se visita j haya tiempo para que el vehículo visite la instalación de disposición (abierta) más cercana a j y luego regrese al depósito antes del final de la jornada laboral.

3.2.3. Operadores de mejora de la solución

La solución inicial se puede mejorar significativamente mediante operaciones simples como reubicación, intercambio y reposicionamiento de clientes entre rutas de vehículos. Después de crear una solución inicial, se utilizan operadores de perturbación de la solución para mejorar la ruta usando búsqueda local con una vecindad o más estructuras de vecindad. Primera mente se buscaron los mejores métodos de vecindad para resolver el VRPTW, una vez que se identificaron se escogieron tres métodos de vecindad para mejorar la solución. La mayoría de las vecindades utilizados suelen ser lo más reportados debido a su desempeño y previamente introducidos en el contexto de diversas heurísticas de construcción y mejora. A continuación, se presentan los operadores de mejora más simples y se pueden encontrar los más complejos encontrados en una revisión para mejorar la solución para el VRPTW en la **Tabla 3.4**. Ejemplos de los vecindads: 2-opt, O-opt, 2-opt *, reubicar, intercambios, etc.

Autores	Año	Solución inicial	Operadores de vecindad	Observación
Desaulniers et al	2014	Descenso de acantilado	2-Opt, 2-Opt*, Or-Opt, Intercambio cruzado, reubicar.	Método de búsqueda vecindad amplia
Wang et al.	2015	Inserción	Or- Opt, 2-Opt*, λ -	Movimientos son

		más barata	intercambio, swap	incrustado en un recocido simulado
Keskin et al.	2016	Algoritmo voraz	Eliminación de ruta aleatoria, Inserción	Se seleccionan de forma dinámica y adaptativa en función de sus actuaciones pasadas
Hiermann et al.	2016	Heurística de inserción	2-Opt, 2-Opt*, Intercambio, Insertar Eliminar si, reubicar.	Método de búsqueda local para intensificar la búsqueda
Alvarez et al.	2017	Heurística constructiva	Intercambio, Insertar Eliminar si, reubicar.	Método de búsqueda local iterada (ILS) y búsqueda vecindad amplia (LNS)

Tabla 3.4. Tabla de métodos de mejorar para la solución del VRPTW

Los métodos de vecindad tradicionales se pueden dividir en dos categorías: dentro de la ruta y entre rutas. Las vecindades en la primera categoría contienen soluciones en las que se cambia una sola ruta con respecto a la solución de referencia, mientras que, en la segunda categoría, contienen soluciones obtenidas al mover clientes entre dos o más rutas. Una encuesta sobre las vecindades aplicadas al VRPTW se puede mostrar en la **tabla 3.5**.

Tabla 3.5. Tabla de principales características de la heurística para VRPTW

AUTOR	AÑO	Intra ruta			Inter ruta				Metaheurística				
		2-Opt	2-Opt*	Or-Opt	Intercambio	λ - Intercambio	Insertar	reubicar	LS	LSA	ILS	SA	GA

Desaulniers et al	2014	✓	✓		✓		✓	✓		✓			✓
Wang et al.	2015		✓	✓	✓	✓							✓
Keskin et al.	2016						✓		✓				
Hiermann et al.	2016						✓	✓	✓				
Alvarez et al.	2017				✓		✓	✓		✓	✓		

A continuación, revisamos los barrios tradicionales que se han utilizado en las metaheurísticas más importantes de la última década en la **Tabla 3.5**.

Para competir con las mejores heurísticas reportadas para resolver el VRPTW, el algoritmo debe hibridarse con algún (algunos) tipo(s) de procedimiento(s) de mejora, dando como resultado un algoritmo híbrido. Para el VRPTW, al hacer esa búsqueda se pretende obtener muchos mejores resultados al reemplazar operadores de intercambios simples (como mover o intercambiar algunos nodos) mediante un procedimiento de búsqueda local. Esto no conduce a una convergencia prematura, gracias a la técnica explicada continuación. Quizás la heurística más conocida para el TSP es la heurística de intercambio de arco 2-OPT.

En esta tesis, se utilizan algoritmos de mejora de vecindad para optimizar localmente una ruta. Entre estos algoritmos, se utilizan mejora entre rutas y otros se implementan para la mejora de la misma ruta. Una ilustración detallada de todos estos algoritmos se muestra a continuación. Se utiliza un operador de vecindad para obtener una nueva solución x de la solución actual x' en la heurística de recocido simulado. Se seleccionan varios operadores de vecindad (entre los que se enumeran a continuación). Cualquier combinación de los operadores enumerados es posible, incluso uno. Entonces, cada vez que un nuevo se necesita la solución x , se elige aleatoriamente un operador de vecindad del conjunto de operadores de vecindad preseleccionados y se aplican una vez a la solución x . El conjunto de operadores preseleccionados se

determina mediante pruebas experimentales que se analizan. Los posibles operadores incluyen:

2-opt: Este algoritmo es aplicado para mejora entre rutas. El algoritmo 2-opt* conecta cada enlace entre dos nodos de una subruta que está rota y conectada con nodos de otra subruta para encontrar una mejora óptima. La complejidad de la vecindad 2-opt es el orden de $O(n^2)$, donde N es el número de nodos.

Or-opt: este algoritmo también se aplica para la mejora dentro de la ruta. La idea es reubicar uno, dos o tres nodos consecutivos de la subruta original con nuevos bordes sin modificar la orientación. La complejidad de Or-opt es $O(N^2)$, donde N es el número de nodos.

3.2.3.1. 2-opt

El procedimiento 2-opt encuentra un recorrido aleatorio inicial sobre el conjunto de clientes (Suthikarnnarunai, 2008). Este recorrido se mejora eliminando dos arcos, invirtiendo uno de los dos caminos resultantes y volver a conectarlos hasta que no haya más se puede hacer una mejora **figura 3.4**. El peor tiempo de ejecución del procedimiento de 2 opciones es $O(n/v)^2$.

Lo que parece ser la respuesta más intuitiva y común para este procedimiento es $O(n!)$. En cada iteración, el algoritmo puede aplicarse a la mayoría de las inversiones $O(n^2)$, pero el número de iteraciones generales está limitado débilmente por $O(n!)$. Ya que al eliminar un cruce, el algoritmo puede, por lo tanto, crear otros nuevos que conduzcan al peor de los casos en que todas las posibilidades son probados.

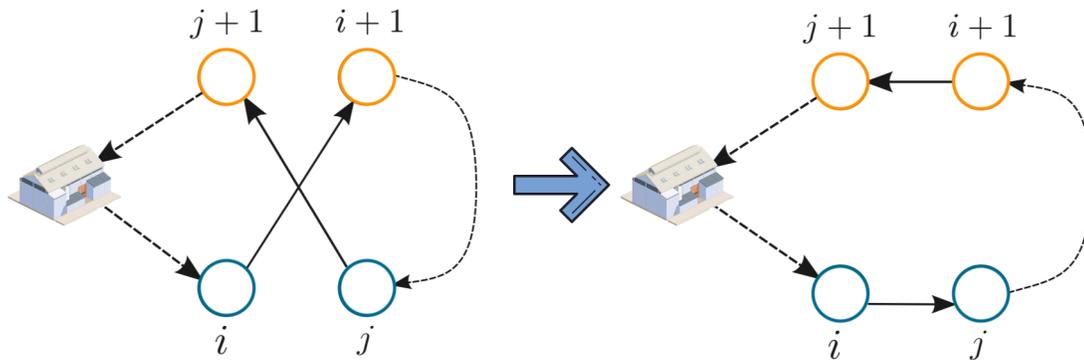


Figura 3.4. Ilustración de la vecindad 2-opt.

Algoritmo 2-opt (Suthikarnnarunai, 2008)

Entrada: Ruta $X_{ijk} = [x_1, x_2, x_3, \dots, x_n]$

Salida: Nueva ruta $X_{ijk} = [x_1, x_2, x_3, \dots, x_n]$

Para $i = 1$ a $n-1$ hacer

 Para $j = 1$ a n hacer

 Si $j < n$ entonces

 Si $d(x_i, x_{i+1}) + d(x_j, x_{i+1}) < d(x_j, x_{i+1}) + d(x_i,$

$x_{j+1})$ entonces

 Intercambiar las posiciones de las dos
 ciudades x_i y x_j

 Fin si

 Sino

 Si $d(x_i, x_{i+1}) + d(x_j, x_{i+1}) < d(x_j, x_{i+1}) + d(x_i,$

$x_{j+1})$ entonces

 Intercambiar las posiciones de las dos
 ciudades x_i y x_n

 Fin si

 Fin si

Fin para

Fin para

Regresar: Nueva ruta

3.2.3.2. Intercambio

El método de intercambio se define de manera similar a la vecindad 2-opt, pero sus soluciones se derivan modificando dos rutas en lugar de una (Meihua, Xuhong & Shumin, 2011). Se eliminan dos arcos, $(i, i + 1)$ y $(j, j + 1)$, de dos rutas distintas, y las rutas se vuelven a conectar insertando los arcos $(i, j + 1)$ y $(j, i + 1)$. Se intercambian las colas de las dos rutas en la **figura 3.5**.

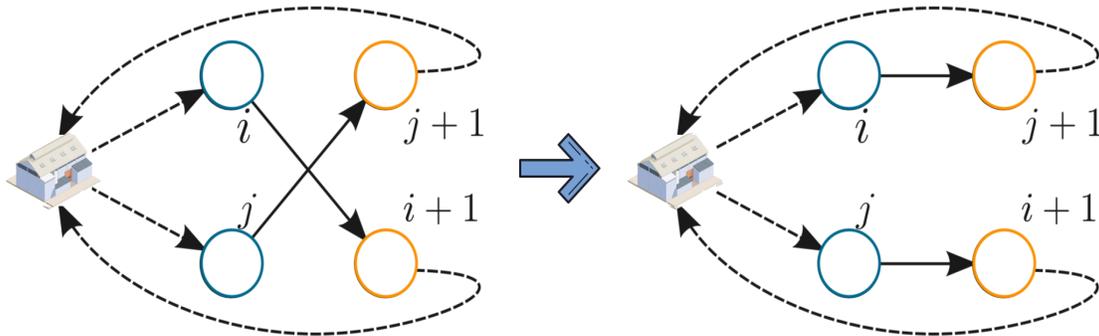


Figura 3.5. Ilustración de la vecindad de intercambio

Algoritmo intercambio (Meihua, Xuhong & Shumin, 2011)

Entrada: Ruta $X_{ijk} = [x_1, x_2, x_3, \dots, x_n]$

Salida: Nueva ruta $X_{ijk} = [x_1, x_2, x_3, \dots, x_n]$

Para $i = 1$ a $n-1$ hacer

 Para $j = 1$ a n hacer

 Si $j < n$ entonces

 Si $d(x_i, x_{i+1}) + d(x_j, x_{i+1}) < d(x_j, x_{i+1}) + d(x_i,$

$x_{j+1})$ entonces

Intercambiar las posiciones de las dos
ciudades x_i y x_j

Fin si

Sino

Si $d(x_i, x_{i+1}) + d(x_j, x_{i+1}) < d(x_j, x_{i+1}) + d(x_i, x_{j+1})$ entonces

Intercambiar las posiciones de las dos
ciudades x_i y x_n

Fin si

Fin si

Fin para

Fin para

Regresar: Nueva ruta

3.2.3.3. Relocalización

La reubicación (consulte la **figura 3.6**), las soluciones se obtienen al reubicar un subtrayecto de una ruta a otro (Balseiro, Loiseau & Ramonet, 2011). Esto se hace eliminando tres arcos $(i, i + 1)$, $(j, j + 1)$ y $(k, k + 1)$ y reemplazándolos por los arcos $(i, j + 1)$, $(j, k + 1)$ y $(k, i + 1)$. Esta método de vecindad puede ser visto como un caso especial de vecindad de intercambio cruzado si permitimos un sub-camino vacío. En el último barrio. La longitud del subtrayecto reubicado suele ser limitada y Algunas implementaciones permiten que la ruta secundaria se invierta cuando se reinserta.

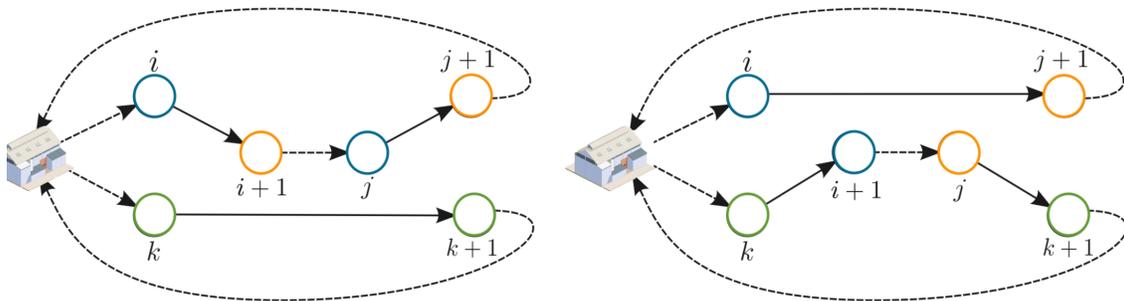


Figura 3.6. Ilustración de la vecindad de relocalización

Algoritmo relocalización (Balseiro, Loiseau & Ramonet, 2011)

Entrada: Ruta $X_{ijk} = [x_1, x_2, x_3, \dots, x_n]$ **Salida:** Nueva ruta $X_{ijk} = [x_1, x_2, x_3, \dots, x_n]$

Generar número aleatorio A1

Generar número aleatorio A2

Mientras $A1 < A2$

Generar número aleatorio A2

Fin Mientras

Para $i = 1$ a $n-1$ hacer Para $j = 1$ a n hacer Si $x_i = A1$ entonces

Agregar A2

Sino Si $x_i = A2$ entonces

Agregar A1

Sino

Agregar nodos restantes

Fin si

Fin si

Fin para

Fin para

Regresar: Nueva ruta

3.2.4. Recocido simulado

El algoritmo de recocido simulado (SA) (Kirkpatrick, 1984) impone diferentes búsquedas aleatorias, aceptación y criterios de paro en el método de búsqueda local para

escapar de los mínimos locales. Los métodos de descenso de búsqueda local no aceptan soluciones que no mejoren la actual solución se mueve en cualquier iteración, mientras que SA lo acepta mediante ciertas probabilidades. Estas probabilidades están determinadas por un parámetro de control (T), llamado temperatura, que tiende a cero.

El interés en el SA comenzó con el trabajo de (Kirkpatrick, 1984), quien propuso un algoritmo SA basado en la analogía entre el proceso de recocido de sólidos y el problema de resolver la optimización combinatoria. Como método probabilístico genérico, simulado recocido tiene capacidad para obtener cerca solución óptima para problemas complejos con gran espacio de búsqueda. SA se ha aplicado con éxito a un gran número de diferentes problemas de optimización combinatoria, incluido el VRPTW (Li & Lim, 2003).

El método de recocido simulado está inspirado en el proceso físico de recocido, donde un material como el acero o el vidrio se calienta y luego se enfría. Simulando el proceso físico; cada paso del algoritmo de recocido simulado reemplaza la solución actual por una solución candidata. La solución candidata es generada al azar utilizando funciones / operadores que son desarrolladas específicamente de acuerdo con el problema. Si la nueva solución (candidata) mejora la función objetivo, entonces acepta el intercambio y se preserva la nueva solución. De otra manera, la nueva solución se acepta con una probabilidad de aceptación, esta aceptación depende tanto de la diferencia entre los valores de la función objetivo y también un parámetro global T (llamado temperatura), el valor de la temperatura va disminuyendo gradualmente durante el proceso. Este mecanismo de recocido simulado suele cambiar la solución aceptar al azar cuando T es grande en iteraciones anteriores. Sin embargo, la solución actual debe ser conducida gradualmente al área óptima a medida que va disminuyendo T. La asignación por aceptar un peor la solución permite el recocido simulado evitando áreas locales óptimas (Kirkpatrick, 1984).

El recocido simulado clásico (Kirkpatrick, 1984) requiere tres principales procesos para resolver el VRPTW de la siguiente manera:

1. Generación de una solución inicial
2. Producir una nueva solución usando operadores de vecindad especiales.
3. Aplicando una función para reducir la probabilidad de aceptar un nuevo peor solución que es útil para escapar de los óptimos locales durante el curso de la búsqueda para el óptimo global

A continuación, se presenta el pseudocódigo del recocido simulado para resolver el VRPTW:

Algoritmo Recocido Simulado (Kirkpatrick, 1984)

Entrada: Ruta $X_{ijk} = [x_1, x_2, x_3, \dots, x_n]$

Salida: Nueva ruta $X_{ijk} = [x_1, x_2, x_3, \dots, x_n]$

Sea $t = \text{temp}0$

Generar solución inicial por el método voraz aleatorio S como solución actual

MIENTRAS $t \geq \text{temp}1$

REPETIR iteraciones Internas

Generar s_j (vecino de s_i) como una nueva solución,
mediante un método de vecindad seleccionado
aleatoriamente

Sea $\Delta Z =$ valor de la resta de función objetivo (distancia)
de $s_i - s_j$

SI $\Delta Z > 0$ ENTONCES

aceptar cambio ($s_i \leftarrow s_j$)

Asignar $S_b \leftarrow s_j$ como la mejor solución

SINO ENTONCES

aceptar el cambio con probabilidad

$$\frac{1}{1 + \exp(\Delta E / T)}$$

FIN SI

FIN REPETIR

Sea $t \leftarrow t * \text{factor de enfriamiento } (\alpha)$

FIN MIENTRAS

Regresar: S_b como la mejor solución

FIN Recocido Simulado

El pseudocódigo de recocido simulado consiste en tres procesos principales que se presentan en el algoritmo. La temperatura t se inicia con el valor de $temp_0$. Iteraciones Internas es una variable que se utiliza para determinar el número de iteraciones para explotar un área de búsqueda local. Esta propiedad habilita el recocido simulado para obtener la mejor solución en el área local y para equilibrar su poder para explotación y exploración en el área de búsqueda. El ciclo de recocido simulado será terminado después de que t llegue a la temperatura final. Una variable llamada factor de enfriamiento se utiliza para disminuir gradualmente valor de la temperatura t . A continuación, se presenta el diagrama de flujo de nuestro algoritmo de recocido simulado en la **figura 3.7**.

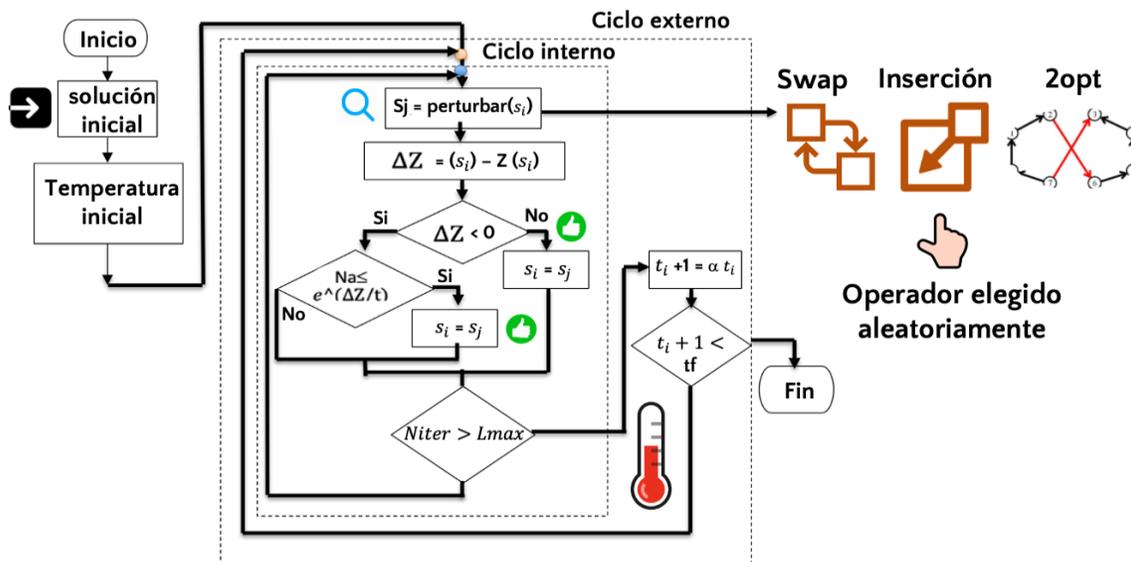


Figura 3.7. Algoritmo de recocido simulado con operadores de vecindad

A continuación, se presenta el diagrama de flujo de nuestro algoritmo de recocido simulado.

3.2.4.1. Parámetros

La clave para encontrar buenas soluciones con el simulado es el esquema de enfriamiento (inicial y final temperaturas y la función para reducir la temperatura) (Alonso-Pecina, 2007). Como otros métodos heurísticos, la configuración de valores para los parámetros de algoritmos es crítica para lograr un buen resultado en razonable cantidad de tiempo. (Alonso-Pecina, 2007) propone utilizar parámetros analíticamente ajustados de recocido simulado. El mejor valor de un el parámetro dado se obtiene cambiando el valor de ese parámetro en cierto rango mientras manteniendo todos los demás parámetros constantes. El algoritmo tiene tres fases, en la primera la longitud de la cadena de Markov (número de iteración del ciclo k metrópolis crece lentamente (de T_0 a T_b) en el segundo, L_k crece con el método de ajuste analítico (de T_b a T_t), y en la tercera fase la cadena de Markov permanece constante (de T_t a T_f).

La temperatura inicial se establece lo suficientemente grande como para aumentar la probabilidad de aceptar nuevas soluciones, sin embargo, se sabe que una temperatura inicial demasiado alta puede causar mucho tiempo de cálculo computacional o un mal desempeño (Park & Kim, 1998). Lleva su tiempo para seleccionar los parámetros del recocido simulado que ofrezca el mejor rendimiento ya que se deben seleccionar los valores de los parámetros. Además, algunos de estos valores de parámetros deben ser determinados simultáneamente debido a la correlación entre ellos.

Un trabajo que resuelve las instancias del VRPTW con el recocido simulado (Mahmudy, 2014), realizó un conjunto de experimentos computacionales que demuestran que la mejora recocido simulado podría producir resultados prometedores en el promedio de cálculo tiempo mediante uso de métodos de vecindad y con la sintonización de los parámetros. Los parámetros que ellos proponen son los siguientes (Mahmudy, 2014): Temperatura inicial 1000; factor de enfriamiento 0.995; temperatura final de 0.01; número de iteraciones 6000. Tomamos en cuenta estos parámetros para nuestro algoritmo, posteriormente se hizo una sintonización de los parámetros mediante

la reducción gradualmente de los valores, pudimos apreciar que a pesar de que temperatura inicial sea alta, el factor de enfriamiento y el número de iteraciones sean muchos esto no asegura que la solución mejore y causa mucho tiempo de cálculo computacional. Es por ello por lo que redujimos un poco los parámetros para poder tener un tiempo de procesamiento razonable y resultados prometedores.



Figura 3.8. Parámetros de recocido simulado basados en (Mahmudy, 2014)

Con base a este conjunto de experimentos preliminares los parámetros para la propuesta simulada el recocido heurístico se estableció como se muestra en la **figura 3.8.**

Tabla 3.6. Tabla de parámetros de recocido simulado

Parámetros	Valor	Descripción
tempInicial	1000	Temperatura inicial del ciclo externo, va disminuyendo gradualmente durante el proceso
α	0.995	Factor de enfriamiento, disminuye gradualmente el valor de la temperatura
tempFinal	0.01	El ciclo de recocido simulado será terminado después de que llegue a la temperatura final
iterInternas	3000	Número de iteraciones fijas por el ciclo interno para explorar un área de búsqueda

		local
--	--	-------

SA se puede terminar cuando la condición de paro alcanza un número determinado de iteraciones que van relacionado a la temperatura es decir cuando la temperatura baja a una temperatura final preseleccionada (Temperatura final).

3.2.4.2. Perturbación de la solución

En este artículo se consideró tres tipos de métodos de vecindad y una combinación de ellos; la idea general es promover la factibilidad aumentando la cantidad de movimientos (NMovimientos) que no violan ninguna restricción. Esta vecindad intenta acercar la nueva solución a la región factible minimizando el costo cada vez que es aplicado. La selección de la vecindad se realiza de la siguiente manera:

Paso 1: Selecciona una vecindad de forma aleatoria,

Paso 2: Realice la perturbación con la vecindad factible (solo los movimientos que no violan las restricciones son considerados). Si NMovimientos = 0, vaya al paso 1.

Paso 3: Si NMovimientos ≥ 1 , haga una selección aleatoria entre los movimientos y realice el movimiento elegido.

El recocido simulado suele usar un mecanismo de perturbación, sin embargo para nuestra implementación se propuso utilizar tres mecanismos de intercambio y una combinación de ellos para generar soluciones vecinas, el método se escoge de forma aleatoria un mecanismo para generar nuevas soluciones **figura 3.9**.

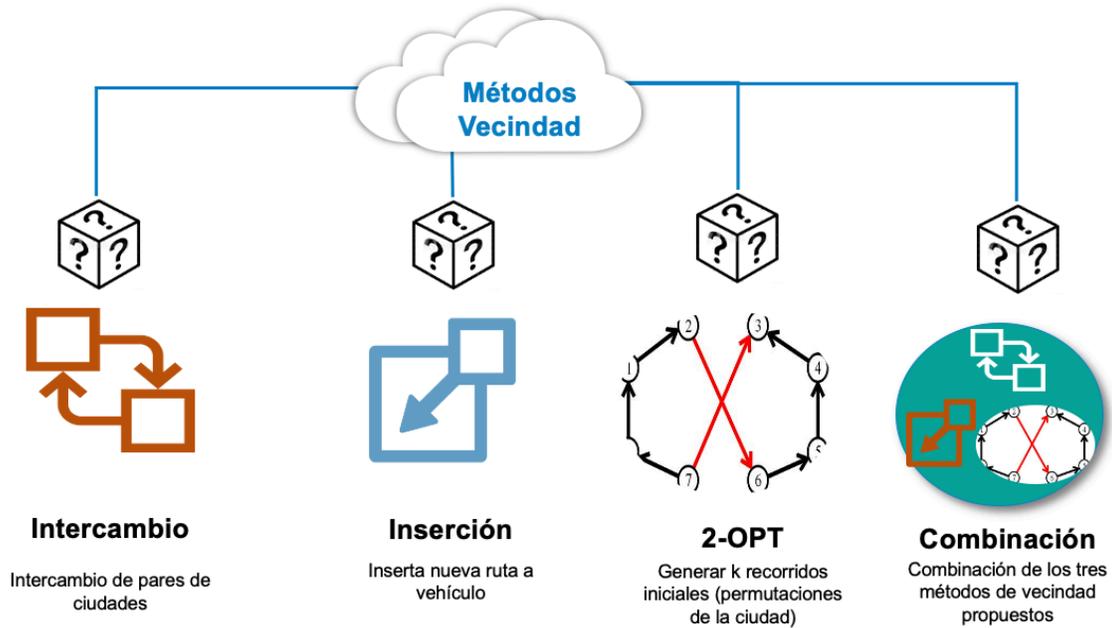


Figura 3.9. Métodos de vecindad para recocido simulado

3.2.4.3. Función de Aceptación

Este esquema de enfriamiento y su implementación contrastan a esquemas clásicos de SA que recurren a la búsqueda aleatoria de vecindad. La mejor solución encontrada, durante la búsqueda se mantiene en lugar de aquella en solución actual cuando el algoritmo se detiene. El algoritmo realiza una única iteración (un intento factible mover) a cada temperatura.

La probabilidad de aceptación se basa en las posibilidades de obtener un nuevo estado con energía E_{k+1} relativo a un estado anterior con energía E_k .

$$h(\Delta E) = \frac{\exp(-E_{k+1}/T)}{\exp(-E_{k+1}/T) + \exp(-E_k/T)} = \frac{1}{1 + \exp(\Delta E/T)} \quad (3.15)$$

$$\sim \exp(-\Delta E/T), \quad (3.16)$$

Donde:

ΔE representa la diferencia de energía entre el presente y los valores anteriores de las energías (consideradas aquí como funciones de costo) apropiadas para el problema físico, es decir, $\Delta E = E_{k+1} - E_k$. Esta es esencialmente la distribución de Boltzmann que contribuye a la función estadística de partición mecánica del sistema.

Se utiliza un valor de k para calcular la probabilidad de aceptar una nueva solución peor. Por lo tanto, determinar un valor apropiado k es crítico. Establecer este valor demasiado bajo causará un azar cambio de la solución actual como la probabilidad de aceptar la peor solución nueva es demasiado alta. Por el contrario, un valor alto de k producirá un valor bajo probabilidad de aceptar la peor solución nueva lo que significa perder la capacidad de escapar de óptimos locales.

3.3. Diseño Experimental

Para probar el desempeño de nuestro algoritmo, se está optando por usar las instancias estándar de VRPTW encontrados en la literatura de VRPTW de (Solomon, 1987) y utilizadas para validar el rendimiento de VRPTW del algoritmo de recocido simulado utilizando diferentes operadores de vecindad. Para las instancias de clientes, se buscará en la literatura una tabla de los mejores resultados y se compararon con los resultados de nuestro algoritmo con los reportados en la literatura. La versión que se informa aquí tiene un objetivo: 1) Minimizar la distancia total de recorrido. Los métodos exactos generalmente usan un objetivo de distancia total monolítica y usan cálculos de distancia y tiempo integrales o de baja precisión (Solomon, 1987). Por lo tanto, los resultados no son directamente comparables con nuestros resultados. Nos comparamos con los mejores resultados obtenidos por heurísticas.

El proceso experimental en el cual nos estamos basando es el de (McGeoch, 2012), los aspectos básicos de los principales experimentos a realizar deben dan respuesta a las siguientes preguntas: ¿Qué debo medir? ¿Qué entradas debo probar? ¿Cómo analizar los datos? Para el proceso experimental, los experimentos se llevan a

cabo en ciclos dentro de ciclos. La planificación de experimentos se alterna con su ejecución. En la etapa de planificación, se formulan preguntas alternas con la construcción de herramientas de prueba y el diseño de experimentos. En la etapa de ejecución, la realización de experimentos se alterna con el análisis de datos. Se pueden realizar pasos individuales en diferentes órdenes y a veces se omiten.

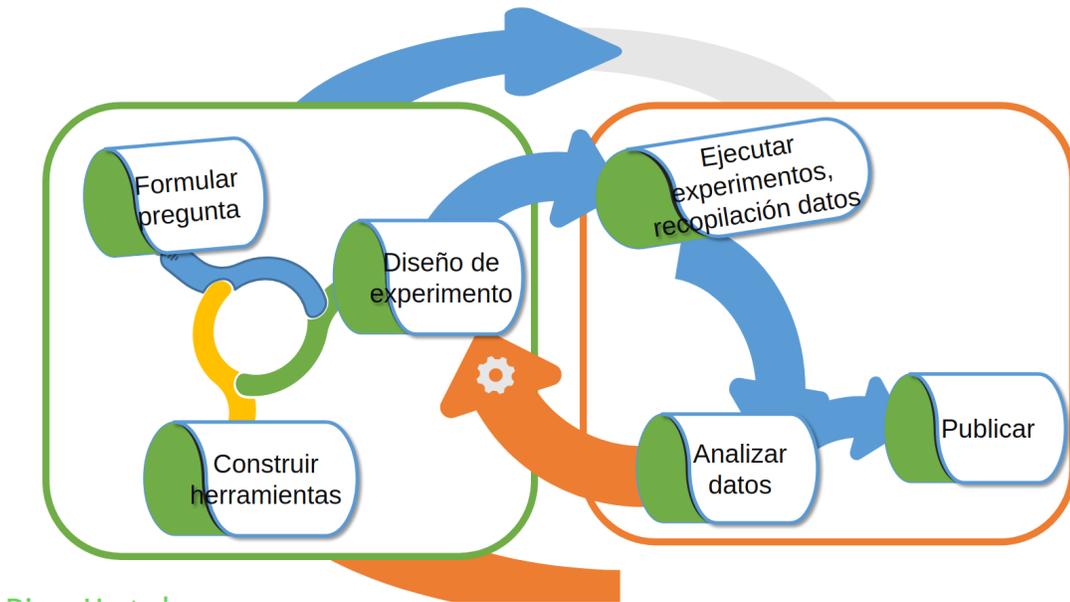


Figura 3.10. Proceso experimental McGeoch, (2012)

Los pasos principales en el proceso experimental se definen en la **figura 3.10**, el proceso no es secuencial, pero sí cíclico: la planificación de experimentos se alterna con la realización de experimentos; los tres pasos en la etapa de planificación pueden llevarse a cabo en cualquier orden; y los pasos pueden saltarse ocasionalmente.

Planifica el experimento según (McGeoch, 2012):

- A. Formular una pregunta
- B. Construir el entorno de prueba. El entorno de prueba comprende el programa de prueba, instancias de entrada y generadores de instancias, herramientas de medición y paquetes, y software de análisis de datos.
- C. Diseño de experimento para abordar la pregunta en cuestión. Se especifica, por ejemplo, qué propiedades se van a medir, qué categorías de entrada se aplican, qué se miden los tamaños de entrada, cuántos ensayos aleatorios se ejecutan, etc.

Ejecución del experimento:

- a. Ejecución de las pruebas y recopilación de los datos.
- b. Aplique el análisis de datos para obtener información y conocimientos. Si la pregunta no fuera respondida, regresar a la etapa de planificación e intente nuevamente.
- c. (Para experimentadores académicos). Publique los resultados. Idealmente, una publicación genera varias preguntas nuevas, que comienzan el proceso nuevamente.

Con base al proceso experimental de McGeoch (2012), se lleva a cabo la implementación de proceso experimental para resolver el problema de enrutamiento de vehículos con el algoritmo de recocido simulado, se utilizan instancias tomadas de la literatura. Por lo cual nuestra primera pregunta fue: ¿El algoritmo de recocido simulado puede resolver instancias de la literatura sobre enrutamiento de vehículos? Para la etapa de experimentación esta etapa sirve para mostrar el procedimiento que se llevará a cabo para producir la solución de las instancias del enrutamiento de vehículos a través del algoritmo de recocido simulado. Nosotros proponemos nuestro proceso experimental para la etapa de la solución de instancias en la **figura 3.11**, la cual consiste en un número de etapas para resolver la problemática.

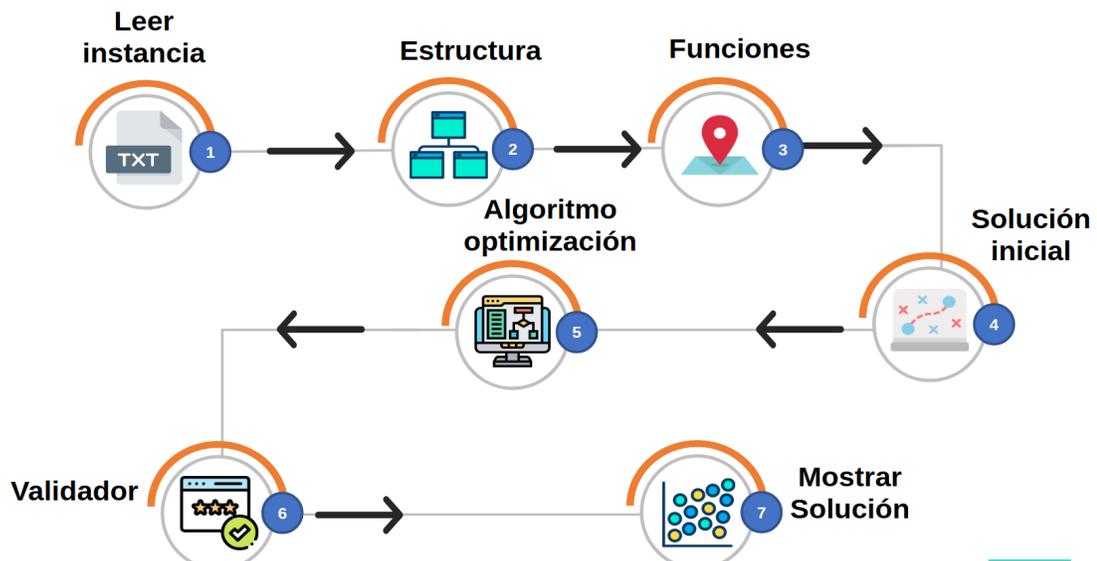


Figura 3.11. Pasos para la resolución de las instancias propuestas.

En cuanto la construcción del ambiente de pruebas, se están utilizando instancias reportadas en la literatura para resolver el problema de enrutamiento de vehículos con

ventanas de tiempo, las primeras instancias que se están tomando son las del VRPTW y posteriormente las instancias reales del sector turístico de Cuernavaca. El lenguaje de programación utilizado fue Python en su versión 3.8. La etapa de la ejecución de los experimentos se realiza con base a las pruebas experimentales realizadas, para ello se toman un conjunto de instancias de prueba del VRPTW, esto para probar la eficiencia del algoritmo de recocido simulado.

El problema de ruteo vehicular con ventanas de tiempo contiene restricciones que deben cumplirse; el número de vehículos, la capacidad de cada vehículo, el número de clientes que debe atender, demandas de cada cliente y el tiempo de servicio que toma atender cada cliente. En la literatura existen instancias del problema de ruteo vehicular, para esta investigación se utilizó el conjunto de datos de instancias de (Solomon, 1987) disponible públicamente. Solomon generó seis conjuntos de problemas. Su diseño destaca varios factores que afectan el comportamiento de los algoritmos de enrutamiento y programación. Datos geográficos; el número de clientes atendidos por un vehículo; porcentaje de clientes con limitaciones de tiempo; y estanqueidad y posicionamiento de las ventanas de tiempo.

En esta sección se encontrarán las definiciones de instancias de clientes de los problemas de referencia VRPTW de Solomon de 1987. La versión que se informa aquí tiene como objetivo: 1) Minimizar el número de vehículos 2) Minimizar la distancia total. La distancia y el tiempo deben calcularse con doble precisión, los resultados de la distancia total se redondean a dos decimales. Los métodos exactos generalmente usan un objetivo de distancia total monolítica y usan cálculos de distancia y tiempo integrales o de baja precisión. Por lo tanto, los resultados no son directamente comparables.

Este conjunto tiene seis diferentes tipos de conjuntos de datos R1, R2, C1, C2, RC1 y RC2 con un tamaño de clientes de 100 (Solomon, 1987). Las coordenadas del cliente son idénticas para todos los problemas dentro de un tipo (es decir, R, C y RC) sin embargo las instancias están clasificadas de dos maneras diferentes, primeramente, el tipo 1 maneja ventanas pequeñas de tiempo y vehículos con capacidades pequeñas, por el contrario, las otras manejan ventanas grandes de tiempo y capacidad grande de los

vehículos. Las coordenadas del cliente siempre son iguales en todos los conjuntos de instancias y solo difieren en el ancho de las ventanas de tiempo, algunos de los conjuntos de datos tienen ventanas de tiempo muy ajustadas.

3.4. Instancias VRPTW

En el VRPTW, cuenta con parámetros, como: C = datos agrupados, R = datos aleatorios, RC = datos aleatorios y agrupados, el valor 1 es para pequeños ventana de tiempo y capacidad de vehículos pequeños y 2 es el gran ventana de tiempo y gran capacidad del vehículo *figura 3.12*.

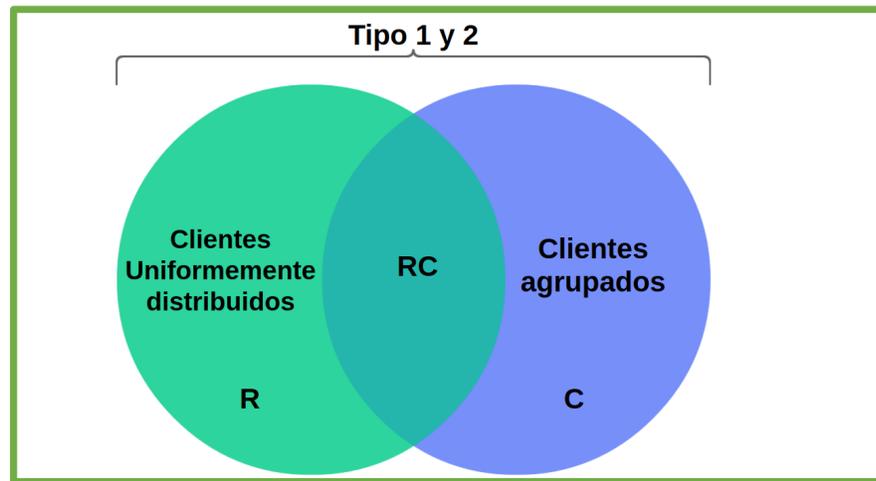


Figura 3.12. Clasificación de las instancias de Solomon para el VRPTW (Solomon, 1987).

Mientras que otros tienen un tiempo muy amplio de ventanas de tiempo representadas aleatoriamente por conjuntos de problemas R1 y R2, y agrupados como conjuntos de problemas C1 y C2. Un tercer conjunto RC1 y RC2 es una mezcla de estructuras aleatorias y agrupadas en el conjunto de problemas. Los datos geográficos se generan aleatoriamente en los conjuntos de problemas R1 y R2, agrupados en los conjuntos de problemas C1 y C2, y una combinación de estructuras aleatorias y agrupadas en conjuntos de problemas por RC1 y RC2. Los conjuntos de problemas R1, C1 y RC1 tienen un horizonte de programación corto y permiten solo unos pocos clientes por ruta (aproximadamente 5 a 10). En contraste, los conjuntos R2, C2 y RC2 tienen un horizonte de programación largo que permite que muchos clientes (más de 30) sean atendidos por el mismo vehículo. Un resumen de estos conjuntos de datos se

encuentra en la **Tabla 4.1**. El objetivo principal de esta tesis es desarrollar buenas metaheurísticas para resolver la recolección de desechos VRPTW, particularmente un problema de referencia de las instancias de Solomon.

Tabla 3.7. Característica principal de las instancias del VRPTW (Solomon, 1987).

Instancia	Clientes	Capacidad del vehículo
R1-tipo	100	200
R2-tipo	100	1000
C1-tipo	100	200
C2-tipo	100	700
RC1-tipo	100	200
RC2-tipo	100	1000

Las características principales de los problemas de prueba se muestran en la **Tabla 3.7**, donde se muestra el tipo de instancia, el número de clientes y la capacidad del vehículo.

Los problemas difieren con respecto al ancho de las ventanas de tiempo. Algunos tienen ventanas de tiempo muy ajustadas, mientras que otras tienen ventanas de tiempo que apenas son limitantes. En términos de densidad de ventana de tiempo, es decir, el porcentaje de clientes con ventanas de tiempo, se creó problemas con 25, 50, 75 y 100% de ventanas de tiempo.

Tabla 3.8. Estructuras de las instancias del VRPTW (Solomon, 1987).

Nombre Instancia	Núm. Vehículos					
	Núm. Cliente	Coord. X	Coord. Y	Demanda	Tiempo inicio	Tiempo fin
Depósito	x_0	x_0	0	E	L	0
Cliente i	x_i	y_i	d_i	a_i	b_i	s_i

En la **Tabla 3.8** se puede apreciar la estructura general de las instancias del VRPTW (Solomon, 1987), la cual contiene el nombre de la instancia, número de vehículos disponibles, el número de cliente (id), la coordenada en X y en Y, la demanda del cliente, la apertura de la ventana del cliente, el cierre de la ventana del cliente y el tiempo de servicio el cual toma para atender al cliente. A continuación, se describe con más detalle el significado de cada una de las variables de las instancias.

X: Coordenada en eje X

Y: Coordenada en eje Y

d_i : Demanda d_i del cliente i

a_i : Inicio de ventana de tiempo en el que el cliente i debe ser atendido

b_i : Tiempo de cierre/fin de ventana de tiempo en el que el cliente debe ser atendido

s_i : Tiempo de servicio = próximo índice más cercano -> tiempo de servicio

Tabla 3.9. Instancia C101 del conjunto de instancias de Solomon (Solomon-benchmark, 2010)

C101						
25						
Núm. Cliente	Coord. X	Coord. Y	Demanda	Tiempo inicio	Tiempo fin	Tiempo Servicio
0	40	50	0	0	1236	0
1	45	68	10	912	967	90
2	45	70	30	825	870	90
3	42	66	10	65	146	90
4	42	68	10	727	782	90
5	42	65	10	15	67	90
6	40	69	20	621	702	90
7	40	66	20	170	225	90
8	38	68	20	255	324	90
9	38	70	10	534	605	90
10	35	66	10	357	410	90

En la **Tabla 3.9** se puede apreciar los primeros valores de la instancia RC101 como forma de ejemplo de la instancia.

El rendimiento de la propuesta de usar recocido simulado se evalúa utilizando un conjunto clásico de instancias para el problema de referencia desarrollados por Solomon para el VRPTW. Recocido simulado es un método estocástico, cada vez que se ejecuta genera una solución diferente por sus métodos aleatorios. El mejor resultado de la simulación el recocido simulado se compara con los mejores resultados producidos por otros enfoques heurísticos que están disponibles en la literatura.

Tabla 3.10. Tabla de mejores resultados de las instancias del VRPTW (Solomon-benchmark, 2010)

Instancia	NV	Distancia	Autores	Instancia	NV	Distancia	Autor
R101	19	1645.79	H	R201	4	1252.37	HG
R102	17	1486.12	RT	R202	3	1191.70	RGP
R103	13	1292.68	LLH	R203	3	939.54	M
R104	9	1007.24	M	R204	2	825.52	BVH
R105	14	1377.11	RT	R205	3	994.42	RGP
R106	12	1251.98	M	R206	3	906.14	SSSD
R107	10	1104.66	S97	R207	2	893.33	BVH
R108	9	960.88	BBB	R208	2	726.75	M
R109	11	1194.73	HG	R209	3	909.16	H
R110	10	1118.59	M	R210	3	939.34	M
R111	10	1096.72	RGP	R211	2	892.71	BVH
R112	9	982.14	GTA	NA	NA	NA	NA
C101	10	828.94	RT	C201	3	591.56	RT
C102	10	828.94	RT	C202	3	591.56	RT

C103	10	828.06	RT	C203	3	591.17	RT
C104	10	824.78	RT	C204	3	590.60	RT
C105	10	828.94	RT	C205	3	588.88	RT
C106	10	828.94	RT	C206	3	588.49	RT
C107	10	828.94	RT	C207	3	588.29	RT
C108	10	828.94	RT	C208	3	588.32	RT
C109	10	828.94	RT	NA	NA	NA	NA
RC101	14	1696.94	TBGGP	RC201	4	1406.91	M
RC102	12	1554.75	TBGGP	RC202	3	1367.09	CC
RC103	11	1261.67	S98	RC203	3	1049.62	CC
RC104	10	1135.48	CLM	RC204	3	798.41	M
RC105	13	1629.44	BBB	RC205	4	1297.19	M
RC106	11	1424.73	BBB	RC206	3	1146.32	H
RC107	11	1230.48	S97	RC207	3	1061.14	BVH
RC108	10	1139.82	TBGGP	RC208	3	828.14	IKMUY

Los resultados experimentales de los mejores resultados se presentan en la **Tabla 3.10**, donde se puede apreciar el nombre de la instancia, el número de vehículos utilizados, la mejor solución en términos de la calidad de la función objetivo y el autor que la resolvió.

Tabla 3.11. Tabla de autores que tuvieron los mejores resultados y método que utilizaron (Solomon-benchmark, 2010)

Año	Acrónimo	Autor	Heurística
2001	BVH	R. Bent and P. Van Hentenryck	Búsqueda local híbrida de dos etapas
2001	BBB	J. Berger, M. Barkaoui and O. Bräysy	Algoritmo genético híbrido paralelo

2002	CC	Z. J. Czech and P. Czarnas	Recocido simulado en paralelo
2000	CLM	J.-F. Cordeau	Heurística de búsqueda tabú unificada
1999	GTA	L. M. Gambardella, E. Taillard, and G. Agazzi	Sistema de colonias de hormigas múltiples
1999	HG	J. Homberger and H. Gehring	Dos metaheurísticas evolutivas
2000	H	J. Homberger	Metaheurística Paralela distribuido
2001	IKMUY	T. Ibaraki, M. Kubo, T. Masuda, T. Uno and M. Yagiura	Algoritmos de búsqueda local efectiva
2001	LLH	H. Li, A. Lim, and J. Huang	Búsqueda local con reinicios similares al recocido simulado
2001	M	D. Mester	Algoritmo de estrategias evolutivas para gran escala
2002	RT	Y. Rochat and E.D. Taillard,	Diversificación e explotación probabilística en la búsqueda local
2004	RGP	L.M. Rousseau, M. Gendreau and G. Pesant	Uso de operadores basados en restricciones
2000	SSSD	G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt and G. Dueck	Registro de resultados de optimización de ruptura utilizando el principio de ruina y recreación
1997	S97	P. Shaw	Un nuevo algoritmo de búsqueda local que proporciona soluciones de alta calidad
1998	S98	P. Shaw	Uso de programación de restricciones y búsqueda local
1997	TBGGP	E. Taillard, P. Badeau, M. Gendreau, F. Geurtin, and J.Y. Potvin	Búsqueda tabú

En la **Tabla 3.11** se puede apreciar los acrónimos de los autores con el método que utilizaron para tener los mejores resultados, se puede apreciar que nuestra aproximación de recocido simulado ha sido utilizada por uno o más autores, como el uso de Búsqueda local con reinicios similares al recocido simulado. Las mejores soluciones en promedio en términos de número de vehículos y distancias totales se obtienen por las heurísticas con respecto a los conjuntos de datos están que están resaltadas en negrita. Las entradas resaltadas comparen el promedio combinado del número mínimo de vehículos seguido de la distancia total recorrido obtenido por la heurística para los conjuntos de datos. La comparación se realiza de esta manera tal como es posible reducir la distancia total recorrida aumentando el número de vehículos.

3.5. Caso de estudio

Una vez que se validó el desempeño del funcionamiento de nuestro algoritmo con instancias de la literatura. Se emplea el algoritmo de enrutamiento de vehículos con ventanas de tiempo para el diseño de rutas de recolección de residuos sólidos en el área del sector turístico en Cuernavaca. Este sector se encuentra el municipio de Cuernavaca, ubicado geográficamente en el norte de Morelos, el cual cuenta con importantes centros turísticos y comerciales, comercios, restaurantes y viviendas, en las que hay un constante movimiento de personas que visitan el Estado de Morelos (SECTUR, 2019). Según el recuento más reciente del Instituto Nacional de Estadística y Geografía, hay un total de 1, 912,211 habitantes, 549 hoteles, 13,271 habitaciones y 2 Pueblos Mágicos. Nos centramos en las áreas del sector turístico y privado de recolección de residuos sólidos. La mayor parte de la composición de los residuos son restos de comida y productos no orgánicos. Elegimos aplicar nuestro algoritmo de enrutamiento de rutas el municipio de Cuernavaca porque juega un papel fundamental en el área del sector turístico. Por lo tanto, es extremadamente importante proporcionar a los ciudadanos un excelente nivel de calidad de vida, que involucra factores que van desde la salud, la vivienda, el transporte, la educación y la riqueza a condiciones naturales; por lo tanto, un aspecto importante de la recolección y transporte de residuos sólidos es la programación operativa y la ruta de los vehículos de recolección involucrados.

Durante el proceso de recolección, los conductores están programados para la recolección diaria de residuos sólidos en sus respectivas áreas operativas. Sin embargo, en el estado actual de la recolección, personas con cierto conocimiento realizan la asignación de los procesos de recolección, transferencia y transporte de manera empírica, con esto los procesos suelen ser afectados por los malos horarios de recolección, los sistemas de recolección de basura inadecuados, la infraestructura insuficiente y el bajo número de vehículos para la recolección de desechos. Las rutas utilizadas por los conductores generalmente se dejan a su discreción y esto se hace sin tener en cuenta la reducción de costos operativos y la conservación del medio ambiente. Esta situación conduce a altos costos de recolección y transporte, así como a la contaminación ambiental. Para una recolección y transporte efectivos de desechos, la calendarización de rutas debe tener en cuenta el enrutamiento sistemático a través de medios científicos que capturen la reducción de costos y la conservación del medio ambiente (Das & Bhattacharyya, 2015).

Uno de los medios para garantizar un rendimiento mejorado en la recolección de residuos sólidos es el enrutamiento efectivo de los camiones de recolección. El enrutamiento representa una ruta entre ubicaciones como el origen y el destino del objeto enrutado. En la recolección de desechos sólidos, el enrutamiento implica la programación y definición de rutas para el recorrido de camiones durante el proceso de recolección. No aplicar métodos científicos o tecnológicos en la selección de rutas por camiones de recolección resulta en sistemas de recolección pobres y costosos (INAPAM, 2017). En la situación actual de la empresa encargada de realizar la recolección no existe una forma sistemática o bien organizada de programar vehículos o camiones para la recolección de residuos sólidos. Por lo general, ésta se basa en la experiencia práctica y métodos intuitivos que resultan en prácticas ineficientes y costosas con implicaciones negativas en la operación comercial y la salud pública, así como en el medio ambiente por lo cual se pretende utilizar algoritmos para optimizar estas rutas de recolección considerando la importancia de las restricciones de la asignación del recorrido.

Este enfoque de la investigación enfocada al enrutamiento óptimo de los camiones de recolección de desechos sólidos analiza la aplicación de diseños de sistemas óptimos a través de programación matemática y algoritmos para mejorar la eficiencia de la recolección. Con el uso de esta técnica se pretende realizar un ahorro del costo total de la gestión de residuos sólidos. Los sistemas inteligentes pueden contribuir significativamente a la optimización de la ruta y ser utilizada como herramienta de apoyo a las decisiones por parte de las empresas de recolección para la gestión eficiente de residuos sólidos en términos de equilibrio de carga dentro de los vehículos *figura 3.13*, gestión del consumo de combustible y generación de horarios de trabajo para trabajadores y vehículos para la minimización general de costos (INAPAM, 2017).

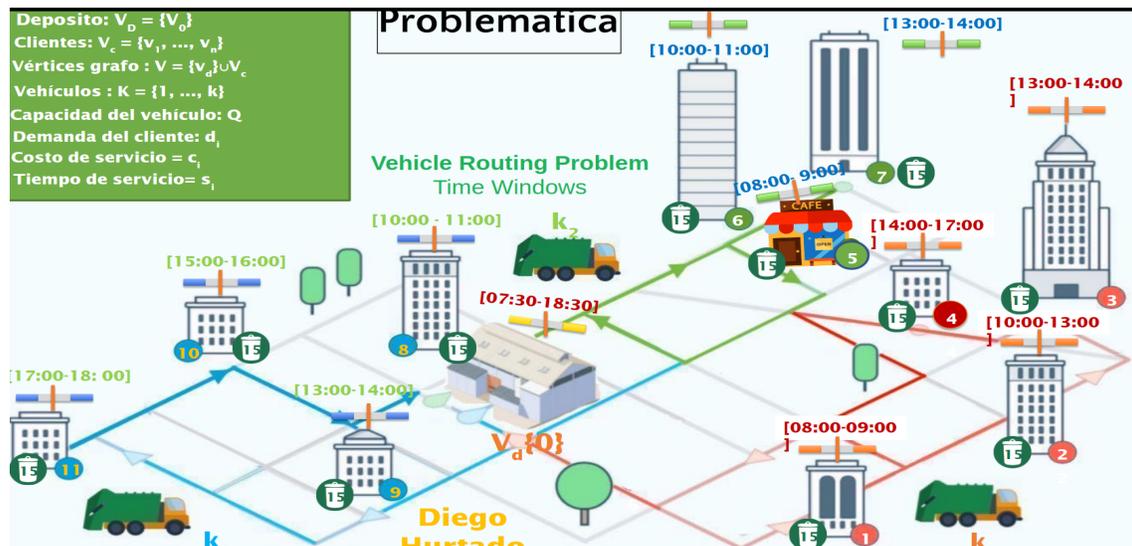


Figura 3.13. Sitios de recolección de residuos sólidos Cuernavaca, Morelos.

En la *figura 3.13* se visualiza un ejemplo de una ciudad que cuenta con un depósito, un conjunto de clientes y cada cliente cuenta con una ventana de tiempo en la cual puede ser atendido, un tiempo de servicio el cual es el tiempo estimado en atender ese cliente, una demanda y un conjunto de vehículos.

A pesar de que la recolección de residuos en general se puede modelar como VRPTW, existen diferentes requisitos que existen en la vida real. En este caso específico de estudio es la empresa que se encarga de hacer la recolección de residuos sólidos en el sector turístico de Cuernavaca, este tipo de soluciones no se ha reportado en la literatura

dentro del sector en México. Para poder resolver las instancias de la vida real se necesitan hacer unas series de adecuaciones que se van a mostrar más adelante con más detalle. Primeramente, se realizó un Pipeline, que nos permite comprender el flujo de trabajo típico para tener la comprensión y la resolución del problema. La **figura 3.14** muestra el Pipeline para abordar nuestra problemática.

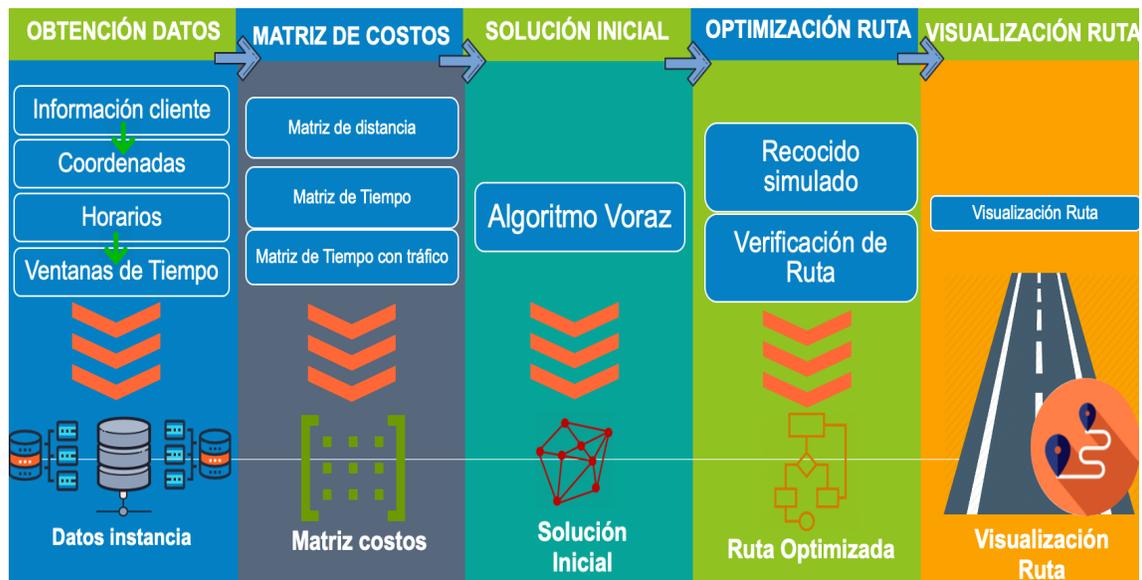


Figura 3.14. Metodología para resolver instancia real

La primera etapa y la etapa que varía más sobre las instancias de la literatura es la obtención de los datos, para obtener la información de los clientes es necesario tener la información de cada uno de los clientes, es decir el nombre del cliente, con ello nos permitirá obtener sus coordenadas del cliente en el mapa real.

En cuanto a las diferencias de una instancia de la literatura, los clientes cuentan con un eje en X y otro en Y para un plano cartesiano, pero ya en la realidad los clientes tienen coordenadas de longitud y latitud en un mapa real. Otra diferencia en cuanto a las distancias de la literatura a las instancias reales son los horarios de las ventanas de tiempo y el tiempo de servicio debido a que en las instancias de la literatura se cuenta con valores enteros de tiempo, es decir que se empieza con un horario enteró que representaría un horario, pero en la vida real se cuentan con horarios del día exactos de hora y minutos. Por último, en cuanto a la demanda de los clientes ésta no varía.

Tabla 3.12. Tabla comparativa de instancias literatura e instancias reales

Número cliente	Coordenada X	Coordenada Y	Demanda	Apertura	Cierre	Tiempo servicio
1	45	50	30	830	1049	90
1	-99.2307500	18.9261000	30	8:00	10:30	30 min

En la **Tabla 3.12**, se presenta un ejemplo más detallado sobre las diferencias entre una instancia de la literatura y una instancia real.

La siguiente diferencia importante es el cálculo de la distancia entre los clientes, usualmente las instancias de la literatura utilizan métodos para calcular la distancia entre dos puntos mediante una línea recta (distancia euclidiana) en un plano cartesiano, pero en la vida real no se puede hacer al cálculo de la distancia entre clientes con líneas rectas debido a se cuenta con calles, avenidas, etc. En la **figura 3.15** se puede apreciar la diferencia.

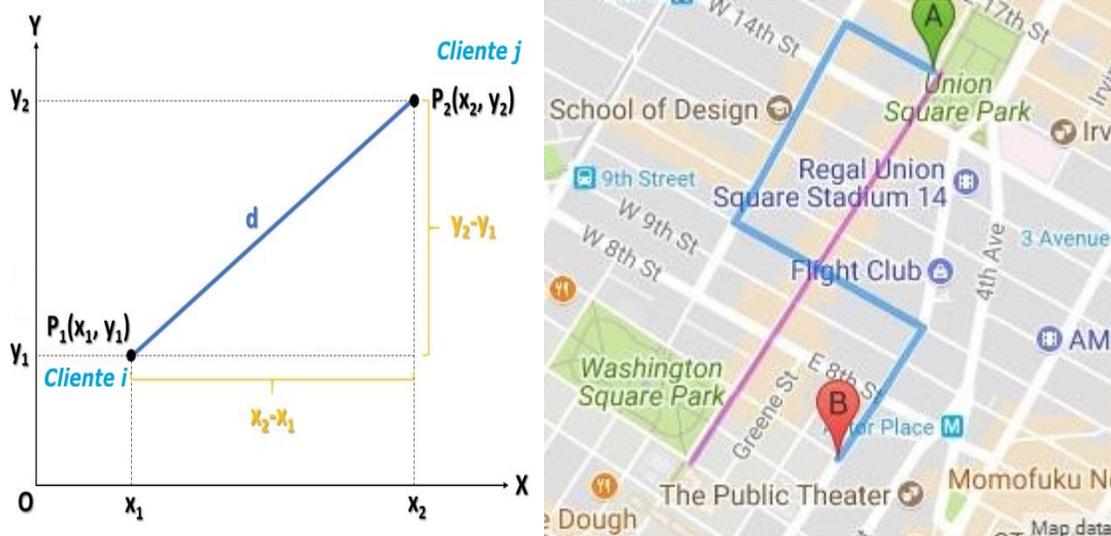


Figura 3.15. Diferencias de distancias de instancias e instancias reales

Para ello se necesita hacer el cálculo de la distancia entre los clientes con otro tipo de técnicas, sin embargo, la mejor forma y que provee más exactitud es utilizar algún tipo

de API que nos permita saber la distancia entre dos puntos, el tiempo de traslado y el tiempo de traslado con tráfico. Para ello estamos utilizando la API de google que nos trae la información exacta sobre la distancia y el tiempo de traslado para ir a un cliente a otro, una vez haciendo la petición de la distancia entre puntos se hace esta consulta para el cliente y sus adyacentes, es decir su determina su matriz de distancia y su matriz de tiempo. Una vez que ya consideramos estos factores, se pueden optimizar las rutas.

3.5.1.Extracción de datos

La presente sección refiere a la recopilación de datos digitalizado de una variedad de fuentes que permitan recolectar la información sobre las localidades de los clientes. En particular, se relaciona con la agregación de datos con la limpieza de los datos extraídos como agregados después de la extracción y borrando algún tipo de imperfecciones al extraer los datos. La obtención de un conjunto de datos de objetos geográficos, direcciones o coordenadas requiere un “Pipeline” estableciendo los pasos que se deben de seguir. Se debe extraer datos clave, obtener los detalles necesarios, visualizar puntos de datos en el mapa y prepararlos para el algoritmo de optimización de rutas. Este caso de estudio incluye 30 clientes que deben ser atendidos, un depósito y 2 camiones disponibles. Cada camión puede recoger una determinada cantidad de basura diaria. Intentamos encontrar una solución óptima para satisfacer la demanda de basura de los clientes, para ello se realizó una encuesta sobre la información real de los pedidos y demandas.

Tabla 3.13. Estructuras de las instancias reales

Instancia						
Núm. Vehículos						
Núm. Cliente	Latitud	Longitud	Demanda	Tiempo inicio	Tiempo fin	Tiempo Servicio
Depósito	x_i	y_i	0	E	L	0
Cliente i	x_i	y_i	d_i	a_i	b_i	s_i

En la **tabla 3.13** se muestra la estructura de la instancia real, la mayor diferencia entre una instancia de la literatura es que se considera coordenadas geoespaciales, es decir longitud y latitud.

X: Coordenada de latitud del cliente

Y: Coordenada longitud del cliente

d_i : Demanda d_i del cliente i

a_i : Inicio de ventana de tiempo en el que el cliente i debe ser atendido

b_i : Tiempo de cierre/fin de ventana de tiempo en el que el cliente debe ser atendido

s_i : Tiempo de servicio = próximo índice más cercano -> tiempo de servicio

La geo codificación es el proceso de convertir direcciones (como una dirección de calle) en coordenadas geográficas (como latitud y longitud), que se pueden usar para colocar marcadores en un mapa o posicionar el mapa y para poder generar nuestra matriz de costos. Por otra parte, la geo codificación inversa es el proceso de convertir coordenadas geográficas en una dirección legible para humanos. Para ello estamos utilizando la API de google para convertir el nombre o las direcciones de los clientes en coordenadas, la API de geo codificación de google es un servicio que proporciona geo codificación y geo codificación inversa de direcciones. Una vez que se obtienen los datos relacionados a los clientes, es decir sus coordenadas, nombre del establecimiento, se visualizan los puntos **figura 3.16** en el mapa para validar que las coordenadas corresponden y una vez hecho esto se preparan los datos para el algoritmo de optimización de rutas. Antes de calcular cualquier distancia, es útil ver las ubicaciones en un mapa. Para hacer eso, se creó marcadores para nuestro conjunto de ciudades. El primer paso es crear el punto de inicio (depósito) y las ubicaciones de los clientes. El punto de partida es la base de operaciones de la ruta y las ubicaciones del museo son destinos. Estas ubicaciones se eligen al azar en el mapa con una distribución bastante uniforme en toda la ciudad para la visualización de las rutas. En la **figura 3.16** se puede apreciar el conjunto de clientes en su correspondiente ubicación en el mapa. Con ello validamos que las coordenadas obtenidas por la API son correctas.

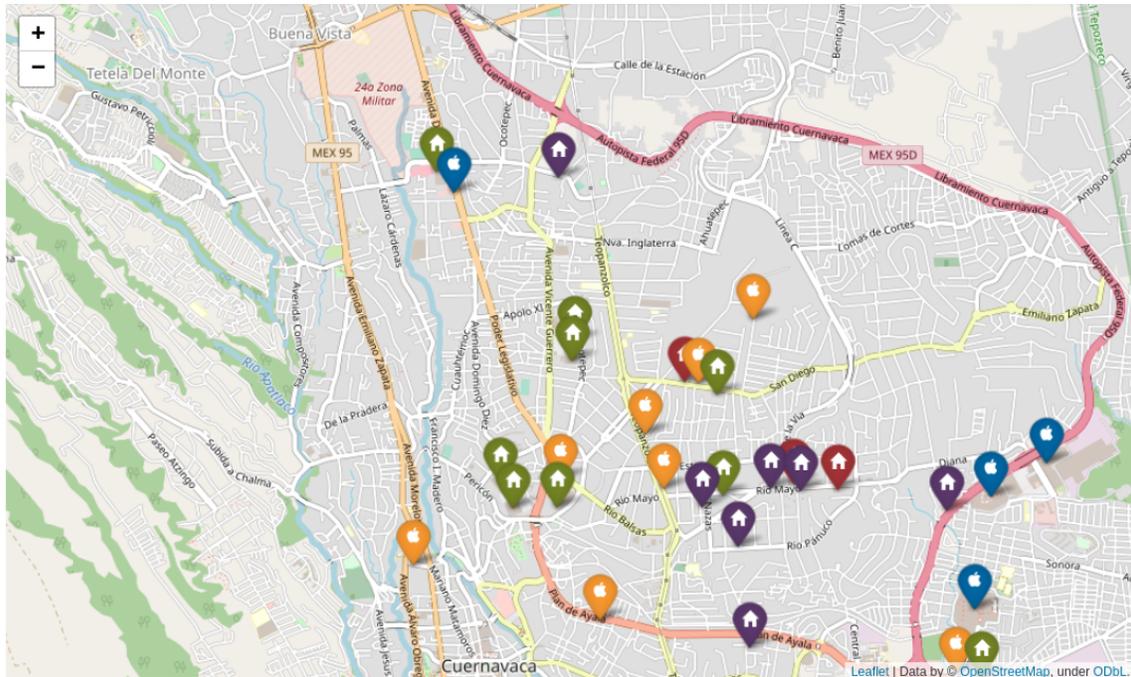


Figura 3.16. Sitios de recolección de residuos sólidos Cuernavaca, Morelos.

3.5.2. Matriz de costos

Debido a que estamos tomando datos reales, no se puede calcular las distancias mediante líneas rectas como lo estábamos haciendo en las instancias euclidianas porque no es un plano cartesiano. Las distancias entre dos puntos sirven para planos cartesianos, pero para mapas reales no, para las aplicaciones actuales reales se tienen necesidades más detalladas como la API de matriz de distancia de google. Con ella se puede calcular qué tan cerca están dos puntos calculando la distancia se está utilizando la API de google de Mapas, la matriz de distancia nos permite obtener la distancia y el tiempo de viaje exacta para una matriz de orígenes y destinos, según la ruta recomendada entre los puntos de inicio y finalización. Mediante la API de matriz de distancia se hacen consultas del tiempo y distancia utilizando orígenes y destinos, junto con una clave personal de la API que nos permite utilizar este servicio. Las consultas de matriz de distancia se están realizando a través de Python, con solicitudes utilizando las coordenadas de orígenes y destinos. El siguiente ejemplo solicita los datos de la matriz de distancia entre Washington, DC y la ciudad de Nueva York, NY, en formato JSON:

```

{
  "destination_addresses" : [ "New York, NY, USA" ],
  "origin_addresses" : [ "Washington, DC, USA" ],
  "rows" : [
    {
      "elements" : [
        {
          "distance" : {
            "text" : "225 mi",
            "value" : 361715
          },
          "duration" : {
            "text" : "3 hours 49 mins",
            "value" : 13725
          },
          "status" : "OK"
        }
      ]
    }
  ],
  "status" : "OK"
}

```

Mediante este tipo de consulta se puede crear una matriz de distancia de los orígenes en cada fila y los destinos en cada columna, lo que le permite recrear una matriz de distancia clásica o usar un origen único con múltiples destinos de distancias y de tiempos. En la **Tabla 3.14** se puede observar un ejemplo de una matriz de adyacencia de distancias.

	Chicago	Detroit	Indianapolis	Milwaukee	St. Louis
Chicago		281	184	92	297
Detroit	283		287	382	530
Indianapolis	183	287		280	242
Milwaukee	92	382	279		373
St. Louis	298	531	244	375	

Tabla 3.14. Ejemplo de matriz de adyacencia de google.

En la tabla se muestra un ejemplo del uso de la API de matriz de distancia incluye un destino (el cliente) y múltiples orígenes (cada técnico potencial). Si estuvieras cerca de Chicago, por ejemplo, podrías ver rápidamente en la primera fila para ver la distancia a Milwaukee (92 millas) o Detroit (281 millas). El patrón diagonal de las celdas vacías es un aviso rápido de que estás viendo una tabla de kilometraje: después de todo, no se puede conducir de Chicago a Chicago.

Antes de comenzar a usar la API de matriz de distancia, se necesita una cuenta de facturación y la API de matriz de distancia habilitada. La API de matriz de distancia desafortunadamente no es gratuita. Tenga en cuenta de facturación antes de utilizar el servicio. La matriz de distancia de un grafo $G = (V, E)$ con n vértices de tamaño $n \times n$ con la propiedad de que $A[i][j]$ tiene asignado el costo de viajar del nodo i al nodo j . En la **Tabla 3.15** se puede observar la matriz de distancia de los clientes, se le asigna el costo de viajar del nodo i al nodo j , este costo está en distancia en metros.

	0	1	2	3	4	5	6	7	8	9	...	22	23	24	25	26	27	28	29	30	31
0	0	11157	11429	6580	6731	8369	10538	11579	11764	16227	...	11271	15869	11639	7695	8074	10439	16651	9344	11537	9693
1	9541	0	1616	3541	3219	1642	1783	1325	2079	3708	...	1558	4037	1384	1857	2438	1684	1944	1392	1508	1125
2	11644	1264	0	5814	3484	2906	812	1580	928	3022	...	271	3351	1650	2122	2154	1102	1362	2077	108	1810
3	6802	4478	5130	0	5518	3323	5492	4719	5346	11288	...	4973	10930	4778	4616	4863	5392	5652	2665	5238	3014
4	6497	3692	4123	5550	0	3172	3769	2858	5051	3592	...	4395	3921	2463	2128	1965	3275	3404	4747	4015	4481
5	6258	2214	2937	3474	2195	0	2584	1899	3865	3755	...	3209	4084	1504	1293	1540	2441	2760	3221	2830	2954
6	9993	452	861	3993	2893	2094	0	1104	1789	3177	...	1132	3506	1059	1531	1678	1024	1284	1844	1076	1577
7	7297	1090	1521	4585	2953	2732	1167	0	2449	2393	...	1792	2722	887	1139	961	1079	1397	2482	1413	2215
8	11860	2192	928	6030	4412	3834	1740	2508	0	3950	...	771	4279	2578	3050	3082	2030	2290	2440	1036	2173
9	16288	4060	3697	10458	4358	4520	3608	3008	4625	0	...	3969	329	3078	3311	2591	2179	2467	4881	3589	4614
10	6876	4222	5399	1809	5593	3397	5566	4794	6900	11362	...	6407	11004	4853	4691	4938	5467	5727	4480	5291	4829
11	7663	3858	4130	1833	10418	4526	5004	4281	4465	8928	...	3973	8570	4340	4812	5394	9571	9352	2045	4238	2394
12	8745	4128	4400	2915	6444	5230	5274	4550	4735	8860	...	4242	8502	4609	5082	5663	5174	9284	2315	4507	2664
13	10091	1526	1797	4261	3841	2628	2672	1948	2161	4331	...	1640	4660	2007	2479	3061	2572	2832	280	1905	61
14	8559	1456	1630	2560	3769	1904	2602	1876	2338	4259	...	1817	4588	1935	2407	2989	2502	2762	1240	1708	973
15	8882	3825	4097	3052	6141	4927	4971	4247	4432	8894	...	3939	8536	4306	4779	5360	9537	9318	2012	4204	2361
16	10319	1297	1569	4489	3613	2400	2443	1719	1932	4103	...	1412	4432	1779	2251	2833	2344	2604	508	1677	241
17	10966	960	1135	3847	3161	1947	1989	1267	1842	3650	...	1322	3979	1326	1799	2380	1889	2149	1155	1212	888
18	7842	2079	3256	1842	3450	1255	3424	2651	3056	5010	...	2535	5339	2710	2548	2795	3324	3584	2137	3149	1871
19	15588	2901	2538	9758	4283	4543	2449	3042	3466	1321	...	2810	963	3112	3486	3308	1964	1745	4293	2431	4026
20	6681	4026	5203	1614	5397	3202	5371	4598	6705	11167	...	6212	10809	4657	4495	4742	5271	5531	4285	5096	4633
21	10691	918	1190	4861	3234	2021	2064	1341	1553	3724	...	1033	4053	1400	1872	2454	1965	2225	880	1298	613
22	11485	1487	223	5655	3707	3129	1035	1803	769	3245	...	0	3574	1873	2345	2377	1325	1585	1918	331	1651
23	17074	4846	4483	11244	4022	5529	4394	4430	5411	1707	...	4755	0	4035	3701	3537	3197	3253	6320	4376	6053
24	6768	1229	1660	4056	1834	2467	1306	395	2588	2778	...	1931	3107	0	472	1054	1448	1708	2307	1552	2041
25	6789	1701	2132	4429	2115	2051	1779	867	3060	2955	...	2404	3284	472	0	163	1642	1960	2757	2025	2490
26	7141	1709	2140	4429	1952	2050	1786	1125	3068	2297	...	2411	2626	730	163	0	1329	1647	3014	2032	2748
27	7998	1462	1099	5004	3118	3104	1010	1458	2028	1914	...	1371	2243	1527	1839	1661	0	479	2854	992	2587
28	8317	1722	1359	5264	3217	3364	1270	1718	2287	1877	...	1631	2304	1787	2158	1980	479	0	3114	1252	2847
29	9804	1805	2077	3973	4121	2908	2951	2227	2440	4611	...	1919	4940	2287	2759	3341	2852	3111	0	2185	341
30	11752	1156	108	4698	3377	2798	704	1473	1036	2914	...	379	3243	1542	2015	2046	994	1254	2548	0	2282
31	10152	1538	1810	4322	3854	2641	2684	1961	2173	4344	...	1653	4673	2020	2492	3074	2585	2845	341	1918	0

Tabla 3.15. Tabla de matriz de adyacencia de clientes.

En la **Tabla 3.16** se puede observar la matriz de tiempo de los clientes, se le asigna el costo en tiempo de viajar del nodo *i* al nodo *j*, este costo está en tiempo en segundos.

	0	1	2	3	4	5	6	7	8	9	...	22	23	24	25	26	27	28	29	30	31
0	0	1130	1206	635	1072	977	1315	1252	1212	1247	...	1215	1218	1205	1273	1306	1327	1319	888	1223	945
1	1142	0	242	537	608	253	257	271	356	604	...	310	680	223	303	406	269	336	177	228	161
2	1225	156	0	658	645	409	105	288	159	497	...	69	573	260	340	421	179	245	312	16	296
3	808	781	856	0	1014	588	927	903	900	886	...	864	857	856	919	949	939	1005	540	872	597
4	1248	686	703	1012	0	720	664	550	862	557	...	773	633	448	417	397	592	629	780	689	764
5	990	441	545	618	426	0	506	455	704	636	...	614	712	353	330	360	499	542	566	531	550
6	1194	51	181	588	561	304	0	244	340	515	...	251	590	176	256	377	184	251	228	175	212
7	1203	245	267	795	561	497	228	0	426	385	...	336	461	201	240	259	249	291	421	252	405
8	1260	309	152	693	797	561	257	441	0	650	...	161	725	412	492	574	332	398	354	169	338
9	1328	633	591	761	676	828	581	563	750	0	...	661	76	534	586	518	389	484	773	577	757
10	775	722	879	277	981	555	894	870	818	853	...	821	824	823	886	916	906	972	494	865	551
11	716	554	630	150	1020	572	717	676	636	671	...	639	642	629	709	812	758	743	312	647	369
12	934	677	753	367	1137	844	840	799	758	795	...	762	766	752	832	935	853	868	435	770	492
13	1014	209	286	448	669	377	373	332	340	666	...	294	741	285	365	467	385	452	30	302	25
14	1018	223	320	413	684	319	387	347	391	681	...	345	756	300	380	482	399	465	155	333	139
15	806	497	573	239	957	664	660	619	578	614	...	582	584	572	652	755	701	686	255	590	312
16	1042	183	260	476	643	350	346	306	314	639	...	268	715	258	338	441	359	425	58	276	42
17	1146	173	270	587	595	303	326	258	341	592	...	295	667	211	291	394	338	405	162	283	146
18	927	425	583	322	685	259	597	574	639	895	...	593	971	527	589	619	610	676	425	568	409
19	1179	450	409	613	753	703	399	569	568	252	...	478	223	540	572	590	340	325	627	395	611
20	826	773	930	328	1032	607	945	922	869	904	...	872	875	874	937	967	957	1024	546	916	603
21	1082	139	215	515	602	310	302	265	270	599	...	224	674	218	298	400	314	381	97	232	81
22	1227	225	69	661	713	478	173	357	161	566	...	0	642	329	409	490	248	314	314	85	298
23	1426	731	689	859	599	943	679	702	848	299	...	759	0	600	569	549	546	582	932	675	916
24	1078	238	255	669	385	432	216	102	414	436	...	325	511	0	80	183	260	326	349	241	333
25	1148	317	334	783	380	491	295	181	493	445	...	404	520	79	0	65	308	350	411	320	395
26	1168	368	390	760	369	468	351	268	549	404	...	459	479	166	51	0	317	359	498	376	482
27	1283	216	175	753	591	469	165	301	334	315	...	244	391	273	320	338	0	119	393	161	377
28	1335	275	233	812	627	528	223	360	392	353	...	303	470	331	371	390	117	0	452	219	436
29	983	239	316	417	699	407	403	362	370	696	...	324	771	315	395	497	415	482	0	332	55
30	1239	140	14	676	628	392	88	272	173	481	...	84	556	243	323	405	163	229	316	0	300
31	1051	244	320	484	704	411	407	367	375	700	...	329	776	319	399	502	420	486	66	337	0

Tabla 3.16. Tabla de matriz de tiempo de clientes.

Nota: Si no se especifica la hora de salida, la elección de la ruta y la duración se basan en la red de carreteras y las condiciones de tráfico promedio independientes del tiempo. Los resultados para una solicitud dada pueden variar con el tiempo debido a cambios en la red de carreteras, condiciones de tráfico promedio actualizado y la naturaleza distribuida del servicio. Los resultados también pueden variar entre rutas casi equivalentes en cualquier momento o frecuencia. Para ello es necesario especificar el tiempo de tráfico en específico que se desea consultar sin embargo si deseara realizar esta consulta en diferentes horarios el costoso el cálculo sería muy costoso debido que para hacer el cálculo de *n* vértices es una matriz bidimensional de tamaño *n* x *n* para cada horario específico, es decir *n* x *n* x *m* donde *m* representa el número de horarios.

Ejemplo: 100 clientes y 6 horas, $100 \times 100 \times 6$, esto sería igual a 60,000 consultas a la API, esto se podría traducir aproximadamente en 800 dólares de consultas de la API.

Nota: Las solicitudes de matriz de distancia que especifican hora de salida cuando modo = conducción están limitadas a un máximo de 100 elementos por solicitud. El número de orígenes multiplicado por el número de destinos define el número de elementos.

3.5.2.1. Parámetros opcionales

Se puede indicar el modo de transporte que desea utilizar. Los modos disponibles son: 'conducir', 'caminar', 'tránsito' o 'andar en bicicleta'. En este caso, la API devolverá los resultados en metros para el modo 'conducir'. Sin embargo, se pueden agregar más parámetros en específico a la consulta como:

Idioma: el idioma en el que devolver los resultados, el código de región, especificado como un valor de dos caracteres ccTLD (dominio de nivel superior de código de país)

Evitar: evitar "peajes", "autopistas" o "ferries" son restricciones a la ruta
el sistema de unidades a utilizar que expresa la distancia

Hora de llegada: especifica la hora de llegada deseada para las solicitudes de tránsito, en segundos desde la medianoche, 1 de enero de 1970, UTC. Puede especificar hora de salida o hora de llegada, pero no ambas.

Hora de salida: la hora de salida deseada. Puede especificar la hora como un entero en segundos desde la medianoche del 1 de enero de 1970 UTC. Alternativamente, puede especificar un valor de ahora, que establece la hora de salida a la hora actual (correcta al segundo más cercano).

Modelo de tráfico: (el valor predeterminado es mejor supuesto): especifica los supuestos que se utilizarán al calcular el tiempo en el tráfico. Esta configuración afecta el valor devuelto en el campo Duración en el tráfico en la respuesta, que contiene el tiempo previsto en el tráfico basado en promedios históricos. Los valores disponibles para este parámetro son:

Mejor supuesto (predeterminado) indica que la duración en transporte devuelta debe ser la mejor estimación del tiempo de viaje dado lo que se sabe sobre las condiciones de

tráfico históricas y el tráfico en vivo. El tráfico en vivo se vuelve más importante cuanto más cerca esté ahora el tiempo de salida.

Pesimista indica que la duración de vuelta en el tráfico debe ser mayor que el tiempo real de viaje en la mayoría de los días, aunque los días ocasionales con condiciones de tráfico particularmente malas pueden exceder este valor.

Optimista indica que la duración en transporte devuelta debe ser más corta que el tiempo real de viaje en la mayoría de los días, aunque los días ocasionales con condiciones de tráfico particularmente buenas pueden ser más rápidos que este valor.

Modo de tránsito: especifica uno o más modos de tránsito preferidos. Este parámetro solo se puede especificar para solicitudes donde el modo es tránsito. El parámetro admite los siguientes argumentos:

- ❖ **bus** indica que la ruta calculada debería preferir viajar en autobús.
- ❖ **metro** indica que la ruta calculada debería preferir viajar en metro.
- ❖ **Tren** indica que la ruta calculada debería preferir viajar en tren.

3.5.3. Ventanas de tiempo de clientes

Muchos problemas de enrutamiento de vehículos implican programar visitas a clientes que solo están disponibles durante períodos de tiempo específicos. Para ello se considera una serie de ventanas de tiempo para los clientes, que son considerados como tiempos solicitados para la visita. Los vehículos deben visitar al conjunto de clientes dentro de su ventana de tiempo. Lo primero que se hizo fue obtener la información sobre los horarios de atención del conjunto de clientes y de los vehículos. Una vez teniendo los datos de las ventanas de tiempo, se pasaron estos horarios a segundos como se muestra en la **figura 3.17** de ejemplo.

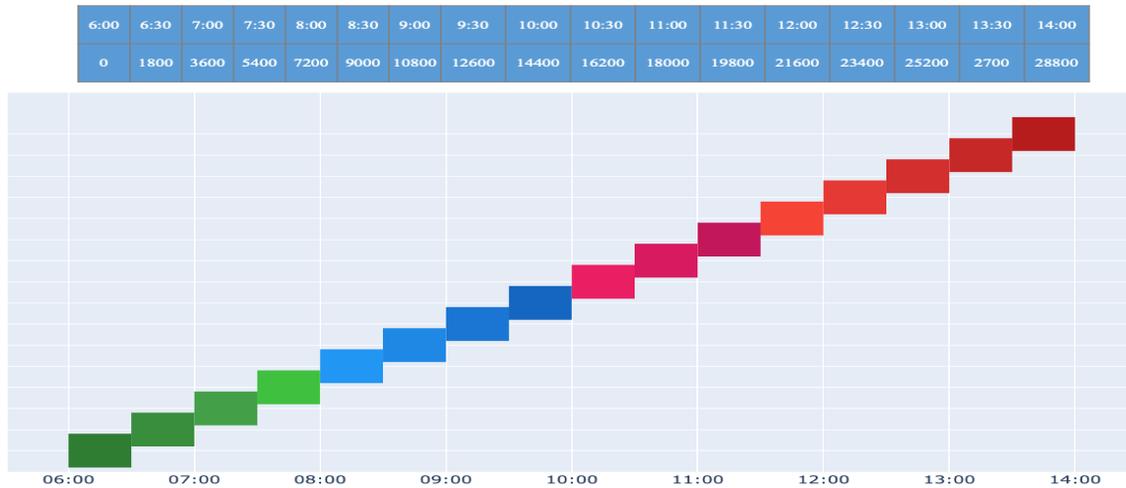


Figura 3.17. Conversión de horarios de atención

A continuación, se muestra un ejemplo de las ventanas de tiempo de un cliente, en este caso se trata de un restaurante que sus horarios de atención son de 6:00 a 8:00, esto se traduce de 0 a 7200 segundos. Debido a que se requiere un tiempo de traslado a ese cliente de 30 min, se le suma al tiempo actual 1800 segundos, posteriormente se le suma el tiempo de servicio de 30 minutos, más tarde se le suma al tiempo actual de 1800 otros 1800 segundos del tiempo de servicio y el tiempo actual es 3600, por lo cual se cumple que esté servicio cae dentro de su ventana de tiempo (véase **figura 3.18**).



Figura 3.18. Ejemplo de ventana de tiempo cliente

3.5.4. Visualización de rutas

La visualización de la ruta en el mapa tiene como propósito mostrar una ruta de viaje junto con instrucciones para seguirlo (ver **figura 3.19**). La ruta en sí misma puede describir un simple viaje a través de la ciudad. Existen numerosos sistemas que pueden proporcionar instrucciones paso a paso para seguir una ruta. Estos se pueden dividir aproximadamente en dos categorías: los que proporcionan una vista animada en tiempo real de la ruta durante el viaje, y los que crean mapas de ruta estáticos. Esta tesis aborda el problema de crear mapas de ruta estáticos adecuados para la recolección de residuos sólidos véase la figura siguiente.

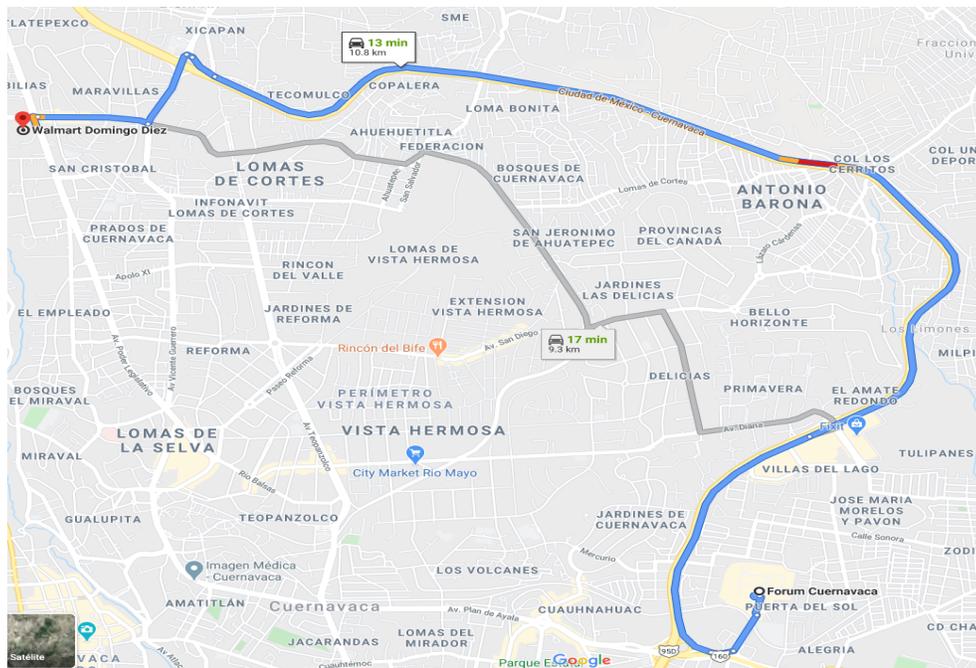


Figura 3.19. Representación de direcciones para ruta

El objetivo de visualizar la ruta es generar una versión imprimible del mapa de la ruta que muestre la vista general y las vistas detalladas de la ruta dentro de un solo marco visual consistente. Nuestra visualización propuesta proporciona un contexto espacial más intuitivo que una simple lista de clientes a seguir. Presentamos una nueva técnica para lograr este objetivo, donde los clientes y las rutas se pueden ver en un mapa. Por lo general, un mapa de ruta proporciona una visión general de toda la ruta junto con instrucciones textuales para seguirlo. Este tipo de mapa de ruta es bastante efectivo, pero

no necesariamente ideal. Considere el escenario de la persona que hace la recolección que imprime un mapa de ruta antes de comenzar un viaje. Es poco probable que el conductor esté dispuesto a memorizar (o incluso leer) las instrucciones de la ruta antes de comenzar el viaje. Al mismo tiempo, las instrucciones serán difíciles de leer mientras conduce. Esto se debe en parte a que una lista de instrucciones textuales contiene pocas señales visuales y el viajero debe escanear la lista visualmente para encontrar la ubicación actual y sus direcciones correspondientes para cada consulta. En la figura se puede apreciar en conjunto de clientes a visitar por el conductor que realiza la recolección véase la **figura 3.20**.

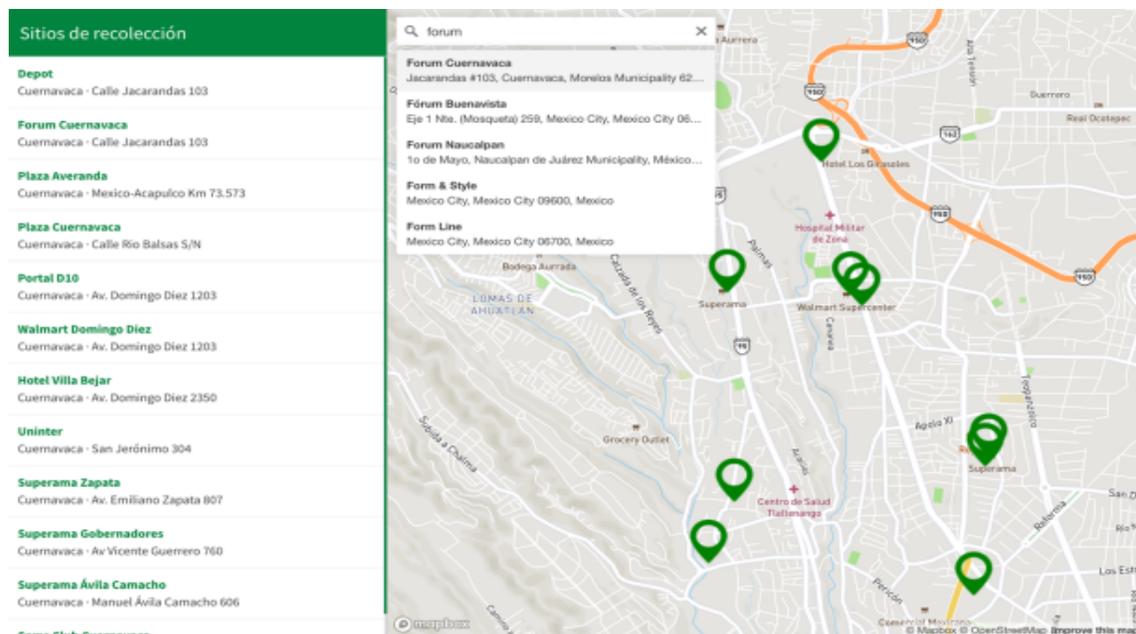


Figura 3.20. Sitios de recolección de residuos sólidos con direcciones

3.5.5. Optimización de rutas mediante API

Las instancias literatura, se utilizaron como base para medir el desempeño de los resultados obtenidos, nos permite evaluar el desempeño del algoritmo y validar el algoritmo. Sin embargo, no sirve para casos reales. Las API de Ruteo proporcionan la distancia y tiempo de viaje real y una ruta optimizada. Sin embargo, se tiene que codificar la solución y formar el modelo del problema, se cuenta con una exploración limitada, sirve para un número pequeño de clientes, no se puede adaptar a cualquier

problemática y tiene un costo asociado por servicio. La solución Ad Hoc, permite crear una solución específicamente para un problema o fin preciso, sin embargo, se tiene que modelar y codificar el problema desde cero. En la **figura 3.21** se puede apreciar un resumen de las diferencias encontradas sobre el uso de las instancias de la literatura e instancias reales.

	<p>Instancias literatura</p> <p>Se utilizan como base para medir el desempeño de los resultados obtenidos.</p>	<p>Ventanas</p> <p>Permite evaluar el desempeño del algoritmo y validar el algoritmo.</p>	<p>Desventajas</p> <p>No sirve para casos reales Específicamente para una problemática</p>	<p>Aportación</p> <p>Permite validar el algoritmo en un entorno controlado</p>
	<p>APIs de Ruteo</p> <p>Proporciona la distancia y el tiempo de viaje de orígenes y destinos.</p>	<p>Ventanas</p> <ul style="list-style-type: none"> -Proporciona la distancia de viaje -Proporciona un ruta optimizada 	<p>Desventajas</p> <ul style="list-style-type: none"> -Se tiene que codificar y formar el modelo del problema -Exploración limitada - Solo sirve para pocos clientes -No se puede adaptar a cualquier problemática - Costo por servicio 	<p>Aportación</p> <ul style="list-style-type: none"> Proporciona la distancia y el tiempo de viaje de orígenes y destinos exactos. Comparación de ruta
	<p>Solución Ad Hoc</p> <p>Solución específicamente elaborada para un problema o fin preciso</p>	<p>Ventanas</p> <p>Permite resolver específicamente un problema.</p>	<p>Desventajas</p> <p>Se tiene que modelar y codificar el problema desde cero.</p>	<p>Aportación</p> <p>Permite resolver específicamente el problema.</p>

Figura 3.21. Comparación de Instancias, API y solución a la medida.

Un factor importante en cuanto optimización de rutas en estudios reales es el uso de APIS de ruteo o soluciones AD HOC, en cuanto el uso de APIS estamos utilizando la API de google que nos trae la información exacta sobre la distancia y el tiempo de traslado de ir a un cliente a otro, una vez haciendo la petición de la distancia entre puntos se hace esta consulta para el cliente y sus adyacentes, es decir su matriz de distancia y su matriz de tiempo. Una vez que ya consideramos estos factores, se pueden optimizar las rutas con su algoritmo de optimización. Pudimos encontrar muchas limitantes al utilizar este servicio, sin embargo, una de las limitantes importantes es que está limitada la búsqueda a un número de clientes, no se puede escalar debido a la limitante de clientes y a que ya se cuenta con un modelo y sus restricciones. Otra limitante es que todo el modelo se tiene que modelar y codificar, se cuenta con sus restricciones del modelo y funciones, pero esto se debe de codificar y adaptar a la problemática.

Capítulo 4. Resultados y discusión

Este capítulo proporciona los resultados del algoritmo propuesto en conocidas instancias de los problemas de referencia VRPTW de (Solomon, 1987) y con las instancias reales de prueba de del sector turístico del caso de prueba.

4.1. Resultados VRPTW

Las instancias del VRPTW de Solomon cuentan con tres tipos de instancias: tipo C datos agrupados, tipo R datos aleatorios y tipo RC datos aleatorios y agrupados, se realizó la ejecución para cada tipo de instancias, posteriormente se muestran los resultados obtenidos en su tabla correspondiente. Las columnas de las tablas muestran el número de vehículos usados mientras que la columna "Distancia" muestra la distancia total recorrida por todos vehículos, el tiempo computacional (en segundos) requerido por el recocido simulado se presenta en la columna "Tiempo". Para medir la calidad de las soluciones obtenidas por el recocido simulado propuesto, la desviación relativa de la distancia total es obtenida por el recocido simulado propuesto con respecto a la mejor solución obtenida en la literatura. Para los parámetros del recocido estamos utilizando los parámetros ya sintonizados previamente sintonizados de nuestro algoritmo para resolver el VRPTW: Temperatura inicial 1000; factor de enfriamiento 0.995; temperatura final de 0.01; número de iteraciones 6000.

El primer experimento se realizó con instancias del tipo C, véase en **Tabla 4.1** , el recocido simulado propuesto produce resultados con el promedio de desviaciones del 13.03% en el conjunto de instancia de tipo C. Los resultados son logrados con el promedio de computacional tiempo de 188 segundos. Por lo tanto, se demuestra la efectividad de los operadores de la vecindad para lidiar con la complejidad del VRPTW para este tipo de instancias.

Tabla 4.1. Tabla de resultados instancias tipo C

Instancia	Autor	NV	NV Heurística	Mejor	Mejor Heurística	Desviación %	Tiempo (s)
C101	RT	10	16	828.94	912.24	9.13	205
C102	RT	10	16	828.94	991.33	16.38	128
C103	RT	10	13	828.06	978.96	15.41	234
C104	RT	10	11	824.78	933.98	11.69	248
C105	RT	10	15	828.94	962.93	13.91	233
C106	RT	10	16	828.94	995.07	16.69	202
C107	RT	10	14	828.94	923.43	10.23	184
C108	RT	10	15	828.94	966.05	14.19	258
C109	RT	10	12	828.94	917.44	9.64	130

En la **Tabla 4.1** se muestran los resultados obtenidos de la experimentación del tipo de instancia C, se puede apreciar que solo en pocos problemas el recocido simulado propuesto produjo desviaciones de más del 15%.

El segundo experimento se realizó con el tipo R, véase en **Tabla 4.2**, el recocido simulado propuesto produce resultados con el promedio de desviaciones del 11.34% en el conjunto de instancia de tipo R. Los resultados se obtuvieron con el promedio de tiempo computacional de 177.667 segundos. Por lo tanto, demuestra la efectividad de los operadores de la vecindad para lidiar con la complejidad del VRPTW para este tipo de instancias.

Tabla 4.2. Tabla de resultados instancias tipo R

Instancia	Autor	NV	NV Heurística	Mejor	Mejor Heurística	Desviación %	Tiempo (s)
R101	H	19	24	1645.79	1732.91	5.02	183
R102	RT	17	22	1486.12	1606.78	7.50	122

R103	LLH	13	22	1292.68	1393.47	7.23	183
R104	M	9	20	1007.31	1221.10	17.50	236
R105	RT	14	22	1377.11	1529.34	9.95	152
R106	M	12	20	1252.03	1415.45	11.54	184
R107	S97	10	18	1104.66	1259.91	12.322	193
R108	BBB	9	19	960.88	1162.09	17.31	152
R109	HG	10	20	1194.73	966.05	8.39	184
R110	M	10	20	1118.59	1258.94	11.12	160
R111	RGP	9	19	1096.72	1258.29	12.84	194
R112	GTA	10	19	982.14	1161.091	15.412	189

En la **tabla 4.2** se muestran los resultados obtenidos de la experimentación de la instancia (R101 - R112), se puede apreciar que solo en pocos problemas el recocido simulado propuesto produjo desviaciones de más del 15%.

El siguiente experimento se realizó con el tipo RC, véase en **Tabla 4.3**, el recocido simulado propuesto produce resultados con el promedio de desviaciones del 12.46% en el conjunto de instancia de tipo RC. Los resultados se obtuvieron con el promedio de tiempo computacional de 234 segundos. Por lo tanto, demuestra la efectividad de los operadores de la vecindad para lidiar con la complejidad del VRPTW para este tipo de instancias.

Tabla 4.3. Tabla de resultados instancias tipo RC

Instancia	Autor	NV	NV Heurística	Mejor	Mejor Heurística	Desviación %	Tiempo (s)
RC101	TBGGP	14	23	1696.94	1889.51	10.19	145
RC102	TBGGP	10	21	1554.75	1670.09	6.90	292

RC103	S98	10	21	1261.67	1533.60	17.73	264
RC104	CLM	10	19	1135.48	1346.79	15.69	302
RC105	BBB	10	23	1629.44	1752.052	6.99	152
RC106	BBB	10	19	1424.73	1559.86	8.66	218
RC107	S97	10	19	1230.48	1467.19	16.13	265
RC108	TBGGP	10	20	1139.82	1380.46	17.43	235

En la **tabla 4.4** se presentan los resultados obtenidos de la experimentación de la instancia RC, se puede apreciar que solo en pocos problemas el recocido simulado propuesto produjo desviaciones por arriba del 15%.

Tabla 4.4. Tabla de resultados obtenidos por tipo de instancias

Tipo de Instancia	Porcentaje en comparación al óptimo global
R101- R112	11.34%
C101- C109	13.03%
RC101- RC108	12.46%

En la **tabla 4.4** se puede apreciar la desviación promedio para cada tipo de instancias. Una vez demostrado la efectividad de nuestro algoritmo, la siguiente etapa es la resolución de las instancias reales, que se van a tratar en el siguiente apartado.

4.2. Resultados caso de estudio: VRPTW

Estamos utilizando la API de Google Maps para comparar nuestros resultados con los de la API, Google Maps es una de las mejores herramientas de planificación de rutas disponibles para escenarios de enrutamiento simples. Crea rutas eficientes para reducir los costos y mejorar la experiencia de los clientes. Tiene un costo y funciones integradas para crear los modelos sin embargo estos se tienen que acoplar al problema particular. A

menudo, Google Mapas se confunde con ser una herramienta de optimización de ruta. Definitivamente se puede usar para planificar una ruta con múltiples paradas. La distinción aquí es que si bien Google Mapas es una herramienta que se puede utilizar para encontrar la ruta más corta entre varias paradas, nunca se diseñó para encontrar el orden óptimo de esas paradas en su ruta. Es por ello por lo que es necesario realizar una solución a la medida del problema. Se están considerando un conjunto de 31 clientes y un depósito. Se cuenta con dos vehículos, a cada vehículo se le asignan un conjunto de clientes con el objetivo de minimizar la distancia de viaje considerando las restricciones de capacidad y de ventanas de tiempo.

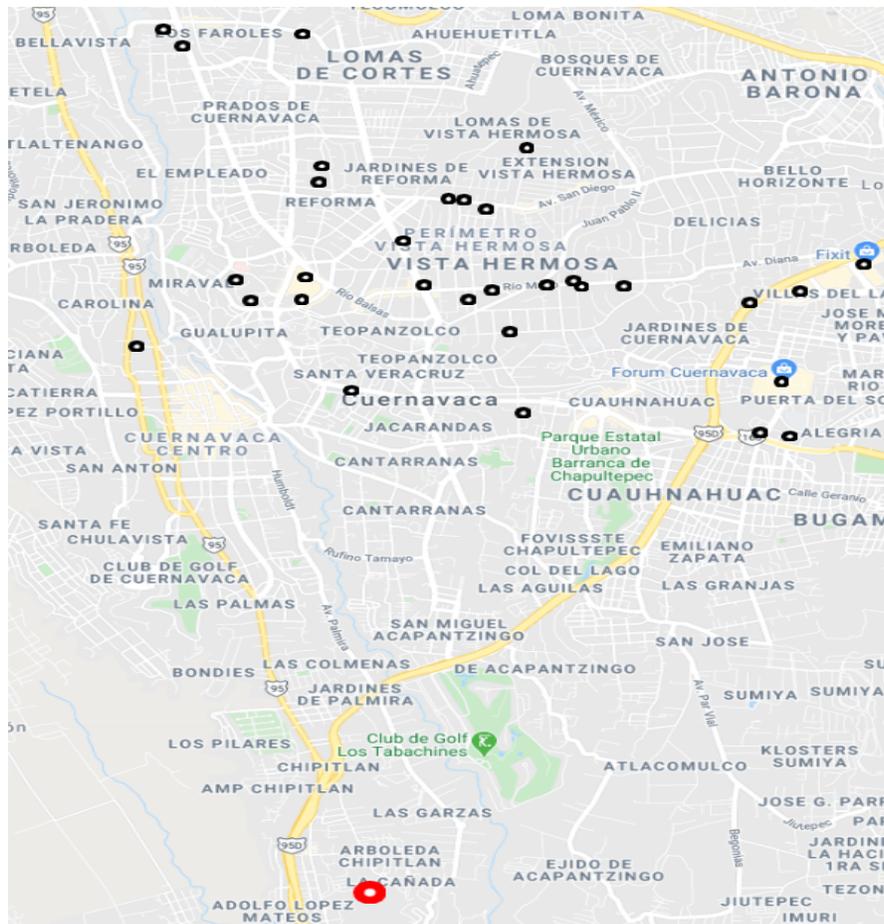


Figura 4.1. Conjunto de clientes a visitar

Para obtener el mejor resultado el recorrido simulado propuesto este es ejecutado 30 veces, se cuenta con dos vehículos con capacidad limitada, el mejor resultado de la simulación el recorrido simulado se compara con el resultado de la ruta obtenida por la

API de google, se compara la distancia de recorrido y se compara la desviación en la **figura 4.2**.

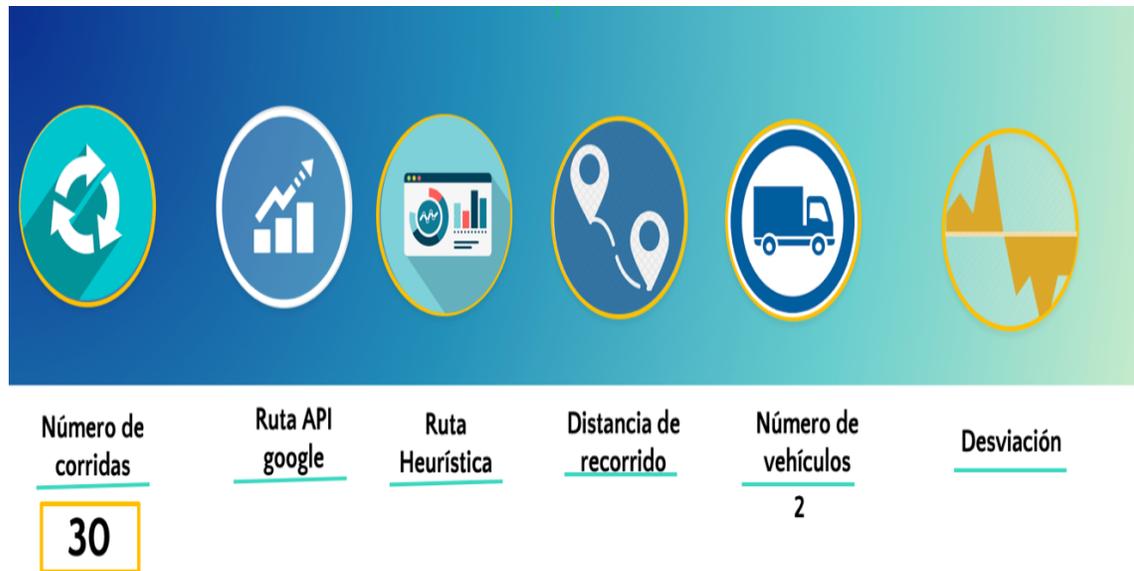


Figura 4.2. Consideraciones para resolver instancias reales

Para medir la calidad de las soluciones obtenidas por el recocido simulado propuesto, la desviación relativa de la distancia total es obtenida por el recocido simulado propuesto con respecto a la mejor solución obtenida por la API de google. Para los parámetros del recocido estamos utilizando los parámetros ya sintonizados: Temperatura inicial 1000; factor de enfriamiento 0.995; temperatura final de 0.01; número de iteraciones 6000. Una vez que se tiene la ruta optimizada, se muestra la primera asignación del vehículo uno con el algoritmo propuesto en la **figura 4.3**, la distancia total es de 26583 metros y 22312.12 segundos.

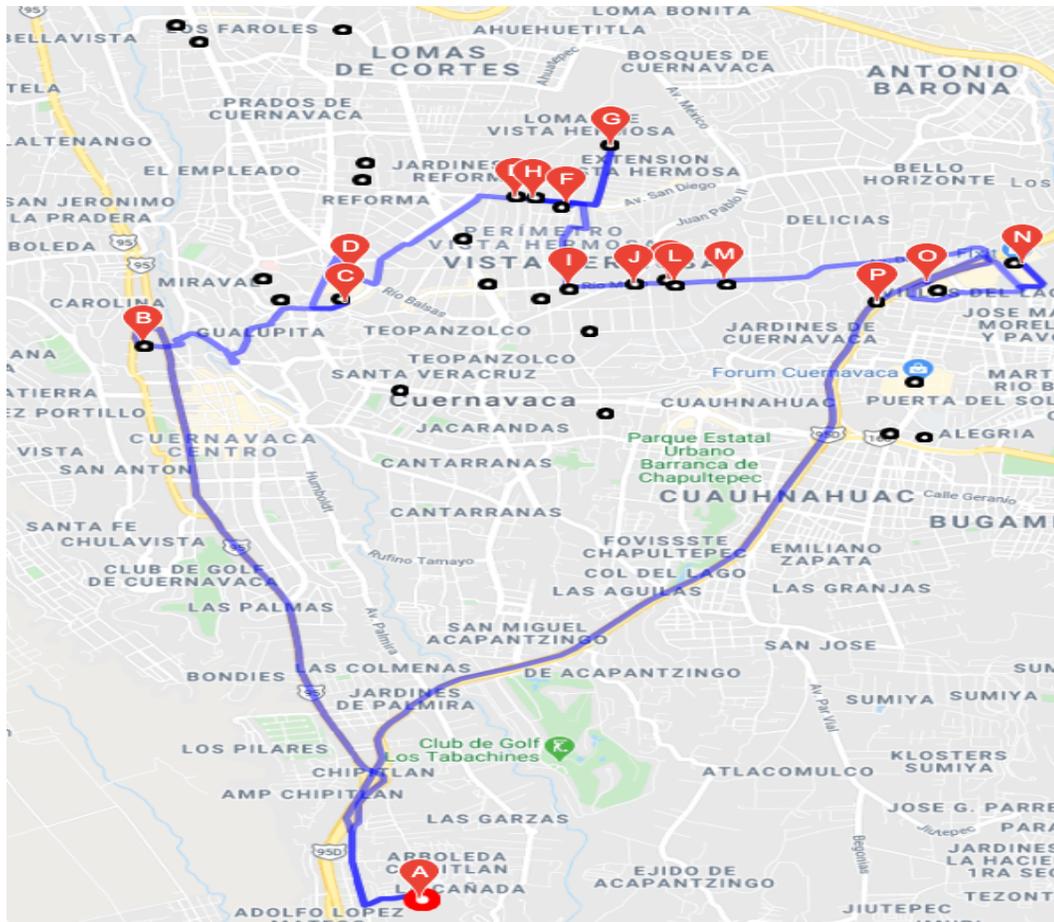


Figura 4.3. Visualización de ruta a) optimizada con algoritmo propuesto

Posteriormente, se muestra la primera asignación del vehículo uno con la API de google en la figura 4.4, la distancia total es de 31,303 metros y 3027.03 segundos.

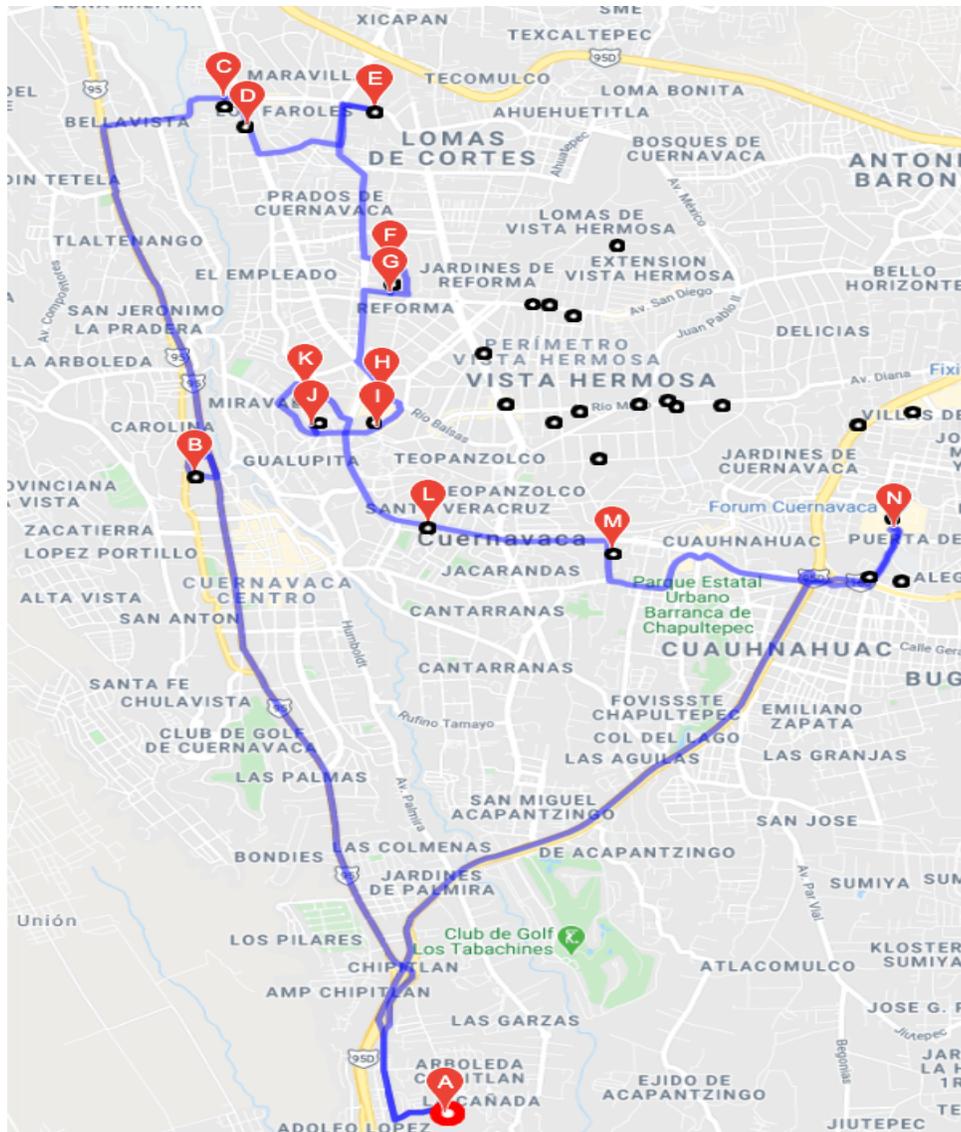


Figura 4.4. Visualización de asignación de ruta a) con API de Google

Posteriormente, se muestra la segunda asignación del vehículo con el algoritmo propuesto en la **figura 4.5**, la distancia total es de 31,423 metros y 18,722 segundos.

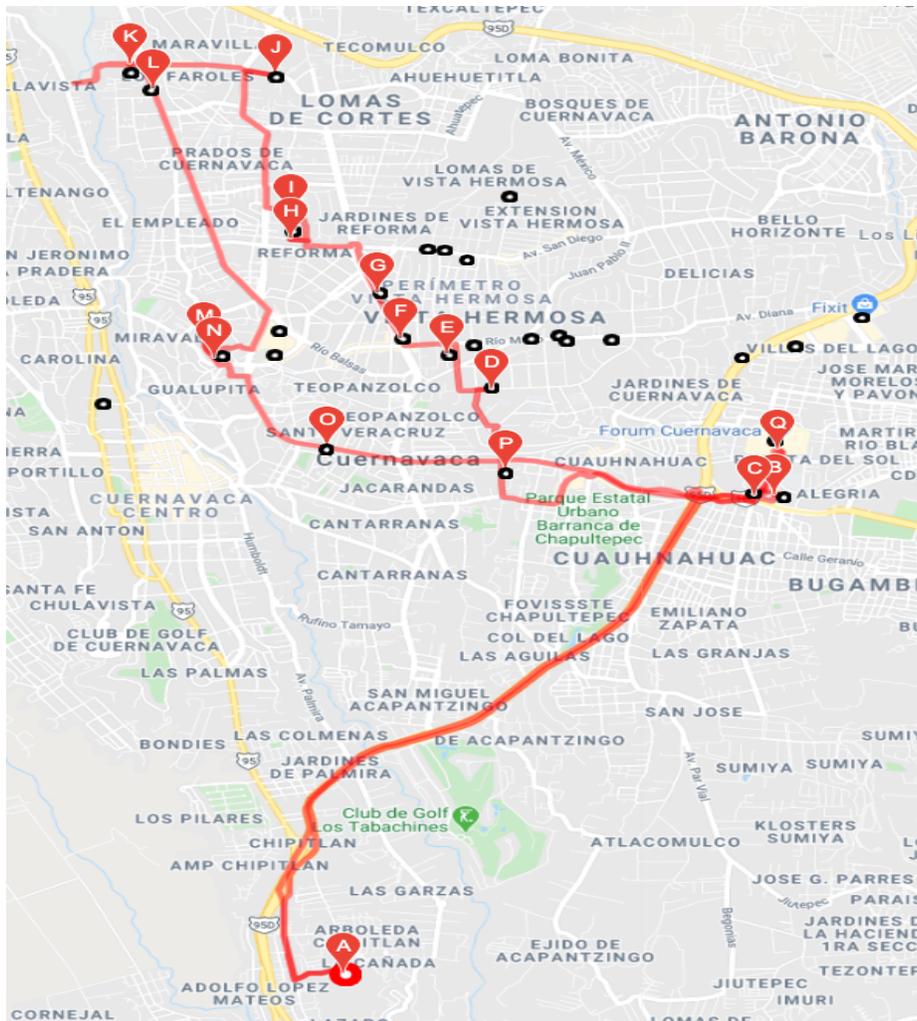


Figura 4.5. Visualización de asignación de segunda ruta b) con algoritmo propuesto

Posteriormente, se muestra la segunda asignación del vehículo con la API de google en la figura 4.6, la distancia total es de 25,931 metros y 17,954 segundos.

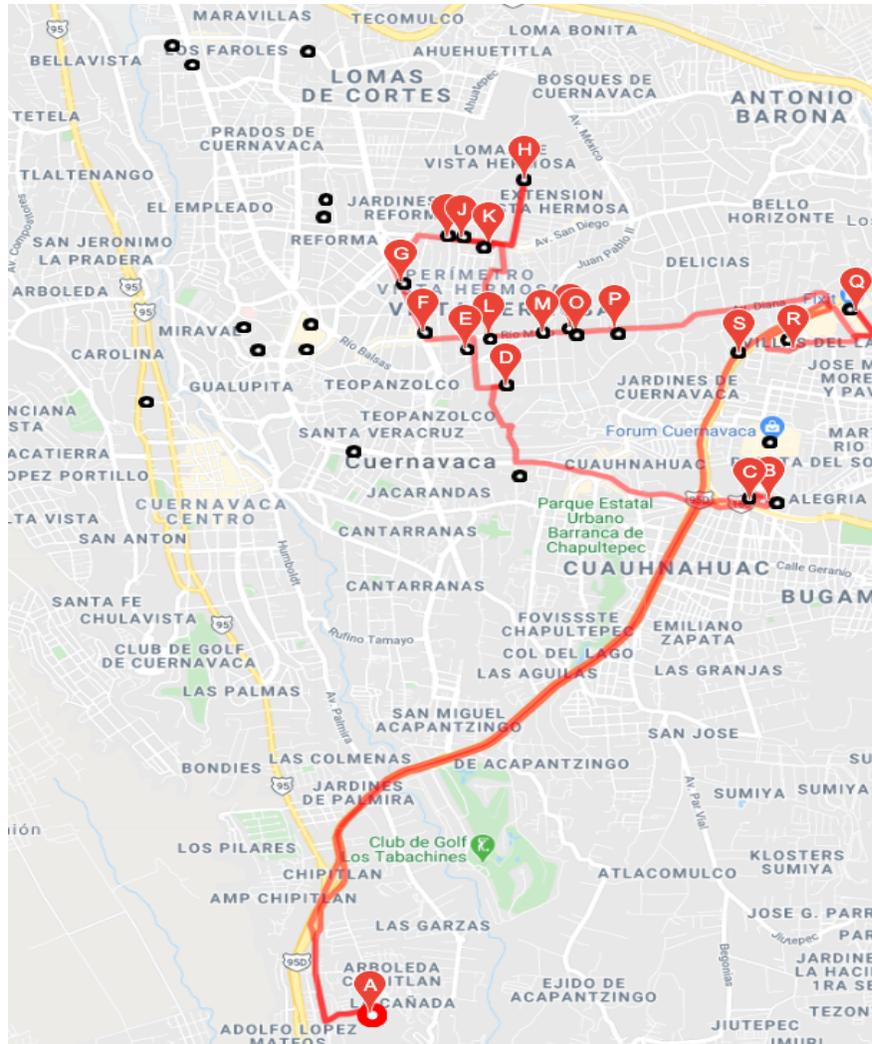


Figura 4.6. Visualización de asignación de segunda ruta b) con API de Google

A continuación, se muestra la ruta completa del vehículo con nuestro algoritmo y con el resultado del algoritmo de la API de google en la **figura 4.7**.

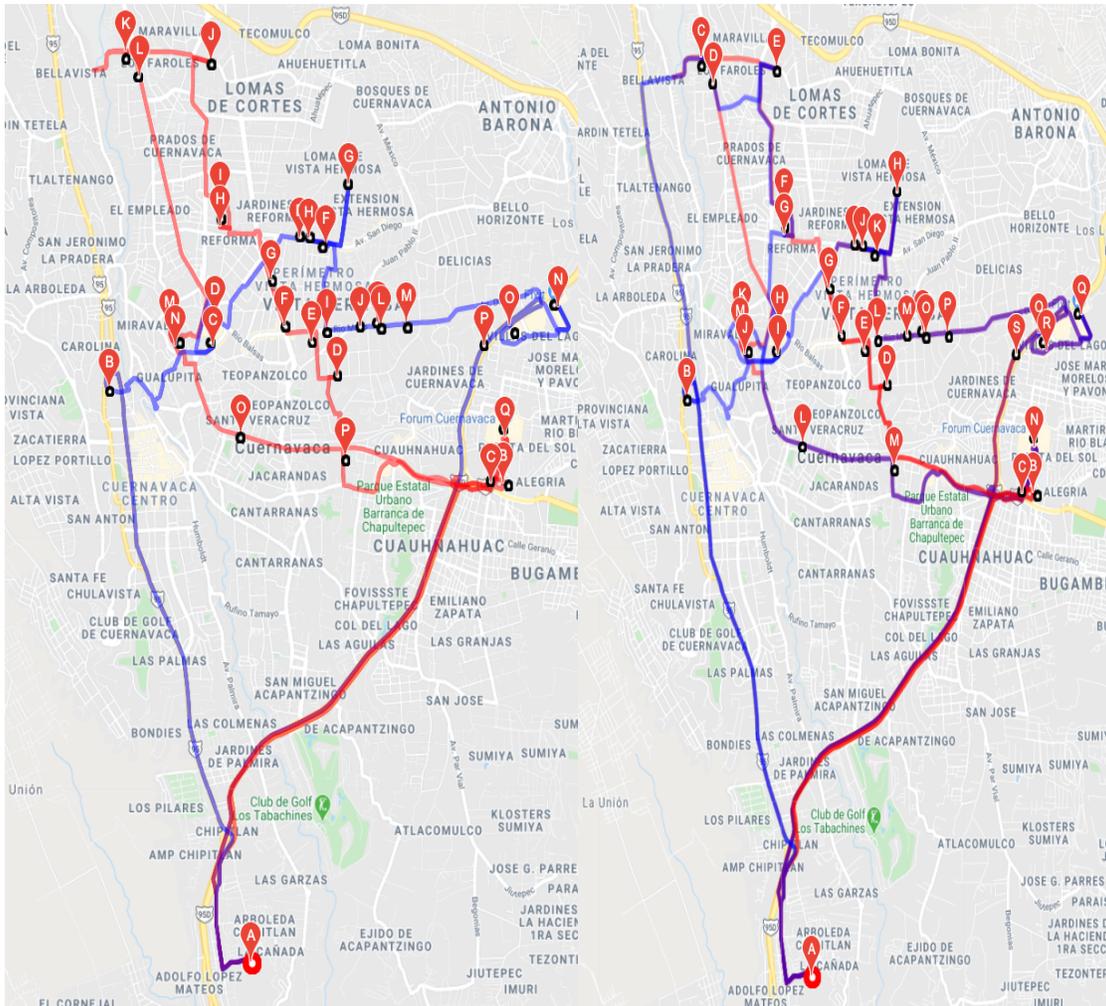


Figura 4.7. Visualización de asignación de rutas con a) algoritmo propuesto y b) API de Google

En la figura 4.7 se puede apreciar a) la distancia obtenida por nuestro algoritmo, con una distancia total de viaje es de 58,006 metros y 22,312 segundos, una demanda de 3,587 kilos para el primer vehículo y una demanda de 4,225 kilos. Con el resultado obtenido por la ruta de google en la figura 4.7 b), la distancia total de viaje es de 57,234 metros y 23,027 segundos, demanda de 4,520 kilos para el primer vehículo y una demanda de 3,292 kilos.

Tabla 4.5. Comparación de ruta obtenida por algoritmo y por API de Google

Algoritmo	Distancia	Tiempo	Demanda vehículo 1	Demanda vehículo 2
-----------	-----------	--------	--------------------	--------------------

Heurística	58,006	22,312	3,587 kilos	4,225 kilos
API Google	57,234	23,027	4,520 kilos	3,292 kilos

En la **Tabla 4.5** se puede apreciar la comparación de la ruta en función de distancia, tiempo de recorrido y demanda de vehículos. Con ello también podemos validar que ninguna restricción se violó.

Posteriormente se realizó el cálculo de la diferencia de utilizar la API de Google y nuestro algoritmo, el cálculo se hizo de la siguiente manera:

$$\frac{\text{Resultado SA} - \text{Mejor Resultado}}{\text{Resultado SA}} 100\%$$

El resultado de la diferencia de nuestro algoritmo al resultado obtenido por la API de google es de -1.3308968037. Lo que podemos apreciar es que la diferencia del error no es mucha y que todas las restricciones de los clientes se cumplen.

Con los resultados del caso de estudio, demostramos que la hipótesis del problema “A través del uso de WCVRP con ventanas de tiempo se puede resolver el problema de recolección de basura en el sector turístico en Cuernavaca; Morelos.” se cumple, nuestra estrategia de diseño produce diseños de rutas estéticamente agradables y visualmente coherentes que tienen menos desorden visual que otras estrategias de diseño, y que constituye un medio eficaz de visualización de rutas.

Capítulo 5. CONCLUSIONES y TRABAJO FUTURO

Este capítulo explicará las conclusiones que se pueden extraer de la investigación realizada y las sugerencias que pueden utilizarse para futuras investigaciones. Basado en la interpretación de los resultados y el análisis realizado en el capítulo, se puede concluir de la siguiente manera.

5.1. Conclusiones

En esta investigación, obtenemos la conceptualización basada en un modelo de enrutamiento de vehículos con ventanas de tiempo para el diseño del servicio de recolección de residuos sólidos en el sector de turismo de Cuernavaca, para resolver la problemática se desarrolló el algoritmo de recocido simulado usando diferentes tipos de operadores para diversificar e intensificar las búsquedas de soluciones tomando en cuenta el marco de referencia de instancias publicadas en la literatura. Para poder resolver el problema, la solución se plantea en dos fases, la primera fase, se obtiene una solución inicial basada en un método voraz estocástico y la segunda fase se aplica recocido simulado con métodos de diversificación de la solución con operadores de vecindad. El algoritmo es capaz de resolver todas las instancias del marco de referencia del VRPTW, teniendo buenos resultados en comparación de los mejores resultados obtenidos en la literatura, con aproximadamente un 12.4 % de desviación del conjunto de prueba. Posteriormente una vez validado el funcionamiento y rendimiento del algoritmo se aplicó para resolver la instancia real mostrando resultados satisfactorios en la optimización de las rutas comparando estos resultados con una herramienta de ruteo de google teniendo como resultado una desviación del -1.33% en comparación de utilizar esta herramienta. Con ello, demostramos nuestra hipótesis “A través del uso de WCVRP con ventanas de tiempo se puede resolver el problema de recolección de basura en el sector turístico en Cuernavaca; Morelos”, en cuanto a la aportación de la

investigación es el diseño de rutas estéticamente agradables y visualmente coherentes que tienen menos desorden visual que otras estrategias de diseño, y que constituye un medio eficaz de visualización de rutas. Por otra parte, un aspecto notable de los resultados es la cobertura del servicio, que garantiza que todos los clientes sean atendidos dentro de sus asignaciones requeridas para generar un mayor control en la planeación de la recolección de los residuos sólidos. La aplicación directa del modelo matemático para la operación de los servicios de recolección sería determinada por los gerentes de servicio; sin embargo, entre los beneficios que se obtendrían, hay mayor eficiencia en el uso de equipos de transporte y cobertura de servicios.

Para el tiempo de realización del proyecto se creó un cronograma de actividades, véase en figura 5.2, para cada semestre con el cual se lleva la asignación de las tareas para realizar la culminación de este, se cumplieron las fechas establecidas para la culminación de este.

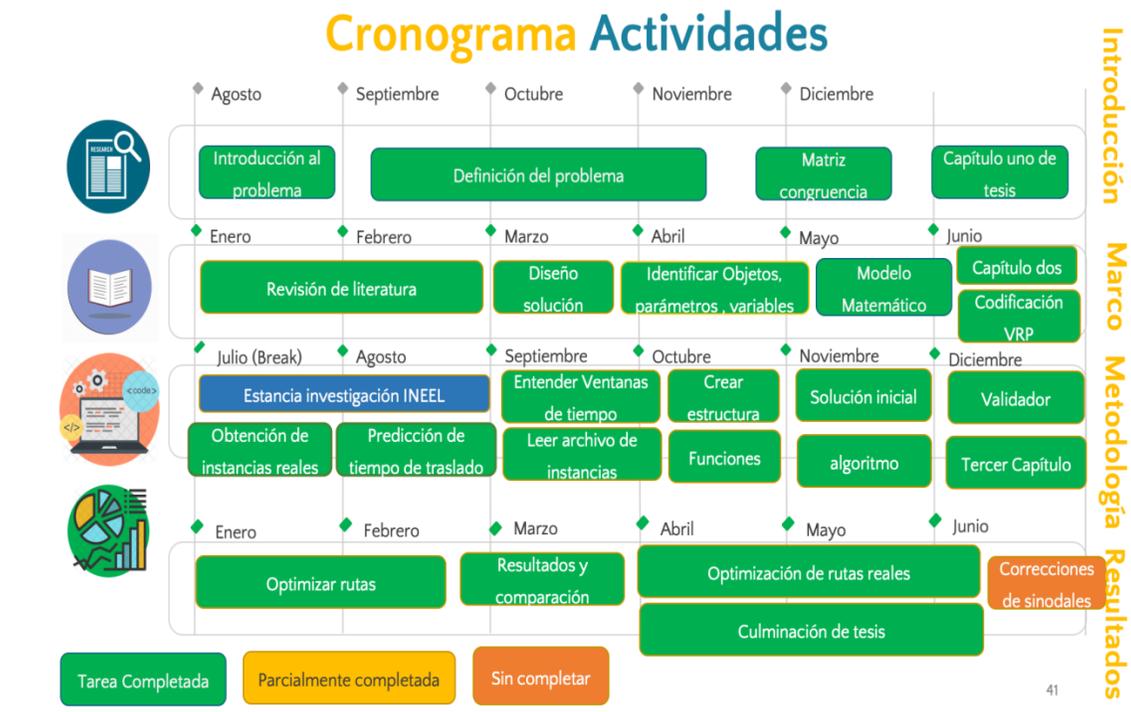


Figura 5.1. Cronograma de actividades del problema de investigación

El proyecto se dividió en 4 etapas (**figura 5.3**), una etapa para cada semestre, la primera etapa consistió en plantear el problema de la investigación definiendo la idea general del problema formalmente, la segunda etapa fue el marco teórico, el cual consistió el hacer la recopilación de los antecedentes, investigaciones previas y consideraciones teóricas en la que se sustenta el problema de investigación, la tercera etapa consistió en la metodología, en la cual se definieron formalmente la solución del problema y la última etapa fue el diseño de experimentos, en el cual se hizo la experimentación de los resultados obtenidos y hacer la prueba de la hipótesis de la investigación. En cada una de las etapas se cumplieron los objetivos de cada una.



Figura 5.2. Fases para realizadas completar el problema de investigación

5.1. Trabajo futuro

Un enrutamiento eficaz para la recolección y el transporte de desechos sólidos es una parte esencial para la sostenibilidad en términos de reducción de costos y conservación del medio ambiente, debido a esto el manejo de desechos sólidos es una prioridad. De la investigación realizada, hay varias sugerencias para futuras investigaciones. Una de las posibles mejoras en cuanto al rendimiento del algoritmo, es la selección de los operados de perturbación de la solución para tratar eficazmente datos agrupados, por ejemplo, agrupando los datos usando un método tales como k-mean de agrupación (Crispín

Zavala, 2015). Para mejorar la eficiencia de los sistemas de enrutamiento de vehículos, los estudios deben centrarse en mejorar la eficiencia mediante la adaptación de los horarios de entrega de las empresas incorporando información relacionada con las condiciones del tráfico en los modelos para evitar generar congestiones de tráfico. Las recomendaciones para estudios adicionales y una mayor eficiencia incluyen implementar un sistema inteligente de gestión de residuos para reducir la cantidad de residuos sólidos, **figura 5.4.**



Figura 5.3. Propuestas de un diseño inteligente de recolección de basura

El sistema intercambia información entre sí mediante redes inalámbricas, y un enrutador y un servidor recopilan y analizan la información para el aprovisionamiento de servicios en ciudades inteligentes: **Contenedores Inteligentes:** los contenedores de residuos inteligentes equipados con sensores de nivel de llenado y detectores con conectividad inalámbrica permiten una planificación, informes y optimización precisos. La red de sensores habilitados en contenedores inteligentes conectados a través de la red celular genera una gran cantidad de datos, que se analizan y visualizan en tiempo real para obtener información sobre el estado de los desechos en la ciudad. Proporciona una visibilidad remota completa sobre cualquier tipo de contenedores y ayuda a planificar las rutas de recolección en consecuencia. El propósito principal del enfoque es el desarrollo de una ciudad inteligente que contenga contenedores inteligentes utilizando dispositivos

del internet de las cosas, los cuales leen el nivel de llenado de un contenedor en tiempo real mediante el uso de sensores y luego activa la compactación automática. Esto aumenta la capacidad del contenedor hasta 5-8 veces. Durante la medición del nivel de llenado, los sensores integrados también miden la temperatura actual dentro de un contenedor y transmiten esta información a un servidor, junto con otros datos medidos. En caso de cambios bruscos de temperatura (por ejemplo, a través de un incendio), el sistema informa automáticamente a los operadores a través de alertas por mensaje de texto o correo electrónico. Supervisa los niveles de plenitud y temperatura de cada contenedor y proporciona alertas de mantenimiento según las solicitudes de los clientes. Los contenedores de basura innovadores vienen equipados con pantallas, paneles u otros sistemas interactivos que captan la atención de los usuarios. Los contenedores de Smart Cities muestran mensajes receptivos para el usuario, activados por sensores en el interior y dependiendo de los tipos de desechos arrojados. Los mensajes se pueden personalizar para audiencias o eventos específicos. Aparte de esto permite a los encargados de hacer la gestión para vigilar los contenedores ubicados en las calles sin vigilarlas constantemente.

Análisis de datos: para una planificación eficaz de la gestión de residuos, se requiere un análisis para producir resultados que puedan facilitar el proceso de planificación. El análisis de datos basado en el aprendizaje sobre los datos de los desechos puede producir una información de generación de residuos más detallada y a tiempo. Con ello se podría llevar a gestionar eficazmente la cantidad de residuos de un área específica. El monitoreo de residuos precisos y oportunos de los datos junto con conocimientos inteligentes basados en datos podrían ser útiles para la planificación de las operaciones de gestión de residuos. **Predicción de tráfico vehicular:** En la vida real, la planificación de rutas tiene mucha incertidumbre, como cambios en la demanda, accidentes de tráfico o condiciones climáticas inesperadas, etc. Todos estos factores se pretenden que se toman en cuenta para que la solución sea aún más apegada a la realidad. El uso de predicción de tráfico de operaciones inteligentes para el transporte proporciona capacidades de predicción de tráfico para las autoridades de transporte de residuos. La función de Predicción de tráfico utiliza los datos de tráfico histórico y en tiempo real recopilados para predecir los niveles de tráfico futuros de un área geográfica, hasta con

una hora de anticipación. El análisis predictivo de la predicción de tráfico realiza cálculos estimados y medios requeridos por el algoritmo para generar niveles de tráfico futuros (**figura 5.5**)

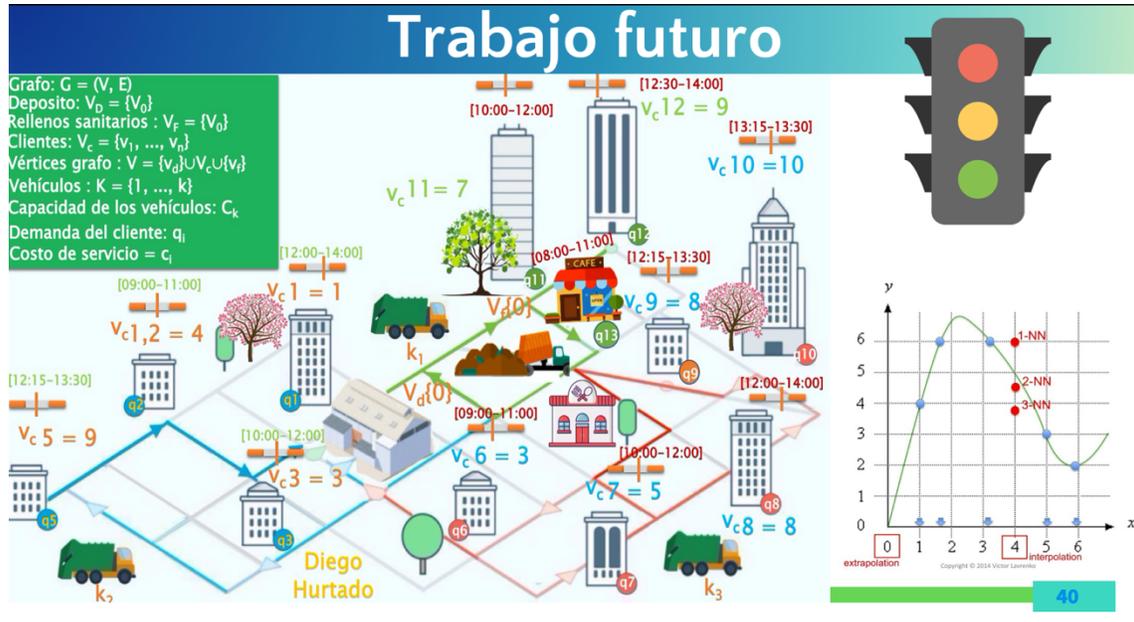


Figura 5.4. Optimización de rutas tomando datos estadísticos de viajes

Los modelos de predicción de generación de residuos convencionales, incluidos los modelos de correlación y regresión, generalmente utilizan factores demográficos y socioeconómicos. Se consideraron varias variables independientes en la mayoría de los modelos de predicción. En el caso de los estudios relacionados con el hogar, se analizó la relación entre los hábitos individuales y la cantidad de desechos, las características individuales de los representantes del hogar en sí. Dependiendo del tipo de casa como un hogar individual, vivienda o una comunidad entera, los tamaños de muestra. la generación se basó en el nivel de densidad de población de un hogar. La mayoría de los investigadores adquirieron datos a través de entrevistas personales y encuestas. La base de datos a nivel individual no estaba disponible debido a cuestiones de censo de protección de datos. Modelo de pronósticos generación de residuos: el objetivo es desarrollar un modelo de pronóstico para determinar la tasa de generación de residuos sólidos urbanos en las ciudades. Este modelo podría permitir a las personas encargadas de las tomas de decisiones de las ciudades determinar las cantidades de residuos generados, operar correctamente los sistemas de gestión de residuos, e incluso adquirir

infraestructura. Visión computacional: La detección de los residuos mediante una cámara la cual capturará una imagen, una vez que se haya reconocido, la cámara capturará su imagen con alguna técnica para detectar las imágenes y predecir que es la imagen, en este caso que tipo de basura hay dentro del contenedor. Cuando el contenedor de basura este lleno, las personas encargadas de la recolección detectarán que el contenedor se encuentra lleno y se enviara la toda la información sobre los residuos dentro del contenedor.

Referencias Bibliográficas

- Alvarez, A., & Munari, P. (2017). An exact hybrid method for the vehicle routing problem with time windows and multiple deliverymen. *Computers & Operations Research*, 83, 1-12.
- Archetti, C., & Speranza, M. G. (2012). Vehicle routing problems with split deliveries. *International transactions in operational research*, 19(1-2), 3-22.
- Balseiro, S. R., Loiseau, I., & Ramonet, J. (2011). An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows. *Computers & Operations Research*, 38(6), 954-966.
- Benjamin, A. M., & Beasley, J. E. (2010). Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. *Computers & Operations Research*, 37(12), 2270-2280.
- Bent, R., & Van Hentenryck, P. (2004). A two-stage hybrid local search for the vehicle routing problem with time windows. *Transportation Science*, 38(4), 515-530.
- Berbeglia, G., Cordeau, J. F., Gribkovskaia, I., & Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. *Top*, 15(1), 1-31.
- Berger, J., & Barkaoui, M. (2004). A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. *Computers & operations research*, 31(12), 2037-2053.

- Berger, J., Barkaoui, M., & Bräysy, O. (2003). A route-directed hybrid genetic approach for the vehicle routing problem with time windows. *INFOR: Information Systems and Operational Research*, 41(2), 179-194.
- Blanton Jr, J. L., & Wainwright, R. L. (1993, June). Multiple vehicle routing with time and capacity constraints using genetic algorithms. In *Proceedings of the 5th International Conference on Genetic Algorithms* (pp. 452-459).
- Bräysy, O., & Gendreau, M. (2005). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation science*, 39(1), 104-118.
- Buhrkal, K., Larsen, A., & Ropke, S. (2012). The waste collection vehicle routing problem with time windows in a city logistics context. *Procedia-Social and Behavioral Sciences*, 39, 241-254.
- Cabrero, E., Orihuela, I., & Ziccardi, A. (2003). Ciudades competitivas-ciudades cooperativas: conceptos claves y construcción de un índice para ciudades mexicanas. *Documento de trabajo*, 139, 32.
- Campbell, A. M., & Wilson, J. H. (2014). Forty years of periodic vehicle routing. *Networks*, 63(1), 2-15.
- Castán, J. A., Ibarra, S., Laria, J., Guzmán, J., & Castán, E. (2014). Control de tráfico basado en agentes inteligentes. *Polibits*, (50), 61-68.
- Chávez, J., Escobar, J., & Echeverri, M. (2016). A multi-objective Pareto ant colony algorithm for the Multi-Depot Vehicle Routing problem with Backhauls. *International Journal of Industrial Engineering Computations*, 7(1), 35-48.
- Chen, H. K., Hsueh, C. F., & Chang, M. S. (2006). The real-time time-dependent vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 42(5), 383-408.
- Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4), 568-581.
- Cordeau, J. F., & Groupe d'études et de recherche en analyse des décisions (Montréal, Québec). (2000). *The VRP with time windows*. Montréal: Groupe d'études et de recherche en analyse des décisions.

- Cordeau, J. F., Desaulniers, G., Desrosiers, J., Solomon, M. M., & Soumis, F. (2002). The vehicle routing problem. Chap. VRP with Time Windows, 157-193.
- Czech, Z. J., & Czarnas, P. (2002, January). Parallel simulated annealing for the vehicle routing problem with time windows. In Proceedings 10th Euromicro workshop on parallel, distributed and network-based processing (pp. 376-383). IEEE.
- Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1), 80-91.
- Dantzig, G., Fulkerson, R., & Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2(4), 393-410.
- Das, S., & Bhattacharyya, B. K. (2015). Optimization of municipal solid waste collection and transportation routes. *Waste Management*, 43, 9-18.
- Das, S., & Bhattacharyya, B. K. (2015). Optimization of municipal solid waste collection and transportation routes. *Waste Management*, 43, 9-18.
- Desaulniers, G., Madsen, O. B., & Ropke, S. (2014). Chapter 5: The vehicle routing problem with time windows. In *Vehicle Routing: Problems, Methods, and Applications*, Second Edition (pp. 119-159). Society for Industrial and Applied Mathematics.
- Díaz-Parra, O., Ruiz-Vanoye, J. A., & Zavala-Díaz, J. C. (2010). Population pre-selection operators used for generating a non-random initial population to solve vehicle routing problem with time windows. *Scientific Research and Essays*, 5(22), 3529-3537.
- Eksioglu, B., Vural, A. V., & Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4), 1472-1483.
- Fisher, M. L., Jörnsten, K. O., & Madsen, O. B. (1997). Vehicle routing with time windows: Two optimization algorithms. *Operations research*, 45(3), 488-492.
- Flood, M. M. (1956). The traveling-salesman problem. *Operations research*, 4(1), 61-75.
- Frausto-Solís, J., Alonso-Pecina, F., & Gonzalez-Segura, C. (2007, December). Analytically tuned parameters of simulated annealing for the timetabling problem. In *WSEAS International Conferences Puerto De La Cruz, Tenerife, Canary Islands, Spain*.
- Fukasawa, R., Longo, H., Lysgaard, J., De Aragão, M. P., Reis, M., Uchoa, E., & Werneck, R. F. (2006). Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical programming*, 106(3), 491-511.

- Gambardella, L. M., Taillard, É., & Agazzi, G. (1999). Macs-vrptw: A multiple colony system for vehicle routing problems with time windows. In *New ideas in optimization*.
- Gendreau, M., Potvin, J. Y., Bräumlaysy, O., Hasle, G., & Løkketangen, A. (2008). Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography. In *The vehicle routing problem: latest advances and new challenges* (pp. 143-169). Springer, Boston, MA.
- Google developers (2020). Vehicle Routing Problem [Online image]. History. <https://developers.google.com/optimization/routing/vrp>
- Grotschel, M., & Lovász, L. (1995). Combinatorial optimization. *Handbook of combinatorics*, 2(1541-1597), 4.
- Gulczynski, D., Golden, B., & Wasil, E. (2011). The multi-depot split delivery vehicle routing problem: An integer programming-based heuristic, new test problems, and computational results. *Computers & Industrial Engineering*, 61(3), 794-804.
- Gutiérrez-Jarpa, G., Desaulniers, G., Laporte, G., & Marianov, V. (2010). A branch-and-price algorithm for the vehicle routing problem with deliveries, selective pickups and time windows. *European Journal of Operational Research*, 206(2), 341-349.
- Hannan, M. A., Akhtar, M., Begum, R. A., Basri, H., Hussain, A., & Scavino, E. (2018). Capacitated vehicle-routing problem model for scheduled solid waste collection and route optimization using PSO algorithm. *Waste management*, 71, 31-41.
- Hiermann, G., Puchinger, J., Ropke, S., & Hartl, R. F. (2016). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, 252(3), 995-1018.
- Hillier, F. S. (2012). *Introduction to operations research*. Tata McGraw-Hill Education.
- Ho, W. K., Ang, J. C., & Lim, A. (2001). A hybrid search algorithm for the vehicle routing problem with time windows. *International journal on artificial intelligence tools*, 10(03), 431-449.
- Holland, J. H., 1975. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, Ann Arbor, MI (reprinted 1992, MIT Press, Cambridge, MA).

- Homberger, J. (2000). Eine verteilt-parallele Metaheuristik. In *Verteilt-parallele Metaheuristiken zur Tourenplanung* (pp. 139-165). Deutscher Universitätsverlag, Wiesbaden.
- Homberger, J., & Gehring, H. (1999). Two evolutionary metaheuristics for the vehicle routing problem with time windows. *INFOR: Information Systems and Operational Research*, 37(3), 297-318.
- Horowitz, E., Sahni, S., & Rajasekaran, S. (1997). *Computer algorithms C++: C++ and pseudocode versions*. Macmillan.
- Ibaraki, T., Kubo, M., Masuda, T., Uno, T., & Yagiura, M. (2001). Effective local search algorithms for the vehicle routing problem with general time window constraints. In *Proc. of MIC (Vol. 2001, pp. 293-297)*.
- Ibaraki, T., Imahori, S., Kubo, M., Masuda, T., Uno, T., & Yagiura, M. (2005). Effective local search algorithms for routing and scheduling problems with general time-window constraints. *Transportation science*, 39(2), 206-232.
- Idrus, Z., Ku-Mahamud, K. R., & Benjamin, A. M. (2017). Waste collection vehicle routing problem benchmark datasets and case studies: A review. *Journal of Theoretical and Applied Information Technology*, 95(5), 1048-1062.
- INAPAM (2017). *MANUAL TÉCNICO SOBRE GENERACIÓN, RECOLECCIÓN Y TRANSFERENCIA DE RESIDUOS SÓLIDOS MUNICIPALES*, SEDESOL, México City, México, 2017. Accessed on: Nov. 02, 2018. [Online]. Available: <http://www.inapam.gob.mx/work/models/SEDESOL/Resource/1592/1/images/ManualTecnicoSobreGeneracionRecoleccion.pdf>
- INE, 2010a. *Potencial de mitigación de gases de efecto invernadero en México al 2020 en el contexto de la cooperación internacional*. 28 de octubre de 2010. México: Semarnat
- Irnich, S., Schneider, M., & Vigo, D. (2014). Chapter 9: Four variants of the vehicle routing problem. In *Vehicle Routing: Problems, Methods, and Applications*, Second Edition (pp. 241-271). Society for Industrial and Applied Mathematics.
- Islam, R., & Rahman, M. S. (2012, May). An ant colony optimization algorithm for waste collection vehicle routing with time windows, driver rest period and multiple disposal facilities. In *2012 International Conference on Informatics, Electronics & Vision (ICIEV)* (pp. 774-779). IEEE.

- Desrosiers, J., Dumas, Y., Solomon, M. M., & Soumis, F. (1995). Time constrained routing and scheduling. *Handbooks in operations research and management science*, 8, 35-139.
- Kara, I., Laporte, G., & Bektas, T. (2004). A note on the lifted Miller–Tucker–Zemlin subtour elimination constraints for the capacitated vehicle routing problem. *European Journal of Operational Research*, 158(3), 793-795.
- Keskin, M., & Çatay, B. (2016). Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 65, 111-127.
- Kim BI, Kim S, Sahoo S. Waste collection vehicle routing problem with time windows. *Computers & Operations Research* 2006; 33: 3624–3642.
- Kirkpatrick, S. (1984). Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics*, 34(5-6), 975-986.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598), 671-680.
- Kohl, N. (1995). Exact methods for time constrained routing and related scheduling problems.
- Kulcar, T. (1996). Optimizing solid waste collection in Brussels. *European Journal of Operational Research*, 90(1), 71-77.
- Kumar, S. N., & Panneerselvam, R. (2012). A survey on the vehicle routing problem and its variants.
- Lageweg, B. J., Lawler, E. L., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1981). Computer aided complexity classification of deterministic scheduling problems. *Stichting Mathematisch Centrum. Mathematische Besliskunde*, (BW 138/81).
- Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European journal of operational research*, 59(3), 345-358.
- Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, 43(4), 408-416.
- Laporte, G., & Nobert, Y. (1987). Exact algorithms for the vehicle routing problem. In *North-Holland Mathematics Studies (Vol. 132, pp. 147-184)*. North-Holland.
- Lenstra, J. K., & Kan, A. R. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 11(2), 221-227.

- Li, H., & Lim, A. (2003). Local search with annealing-like restarts to solve the VRPTW. *European journal of operational research*, 150(1), 115-127.
- Li, J. Q., Borenstein, D., & Mirchandani, P. B. (2008). Truck scheduling for solid waste collection in the City of Porto Alegre, Brazil. *Omega*, 36(6), 1133-1149.
- Lim, A., & Wang, F. (2005). Multi-depot vehicle routing problem: A one-stage approach. *IEEE transactions on Automation Science and Engineering*, 2(4), 397-402.
- López Doriga, 2015. <https://lopezdoriga.com/nacional/el-2015-registra-el-mayor-numero-de-dias-contaminados-en-el-d-f/>
- Mahmudy, W. F. (2014). Improved simulated annealing for optimization of vehicle routing problem with time windows (VRPTW). *Kursor*, 7(3).
- Markov, I., Varone, S., & Bierlaire, M. (2016). Integrating a heterogeneous fixed fleet and a flexible assignment of destination depots in the waste collection VRP with intermediate facilities. *Transportation research part B: methodological*, 84, 256-273.
- McGeoch, C. C. (2012). *A guide to experimental algorithmics*. Cambridge University Press.
- Meihua, W., Xuhong, T., & Shumin, W. (2011). Hybrid ant colony optimization algorithm for two echelon vehicle routing problem. *Procedia Engineering*, 15, 3361-3365.
- Mester, D. A. V. I. D. (2002). An evolutionary strategies algorithm for large scale vehicle routing problem with capacitate and time windows restrictions. In *proceedings of the conference on mathematical and population genetics*, University of Haifa, Israel.
- Mohanty, S. P., Choppali, U., & Kougianos, E. (2016). Everything you wanted to know about smart cities: The internet of things is the backbone. *IEEE Consumer Electronics Magazine*, 5(3), 60-70.
- Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F., & Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79, 115-129.
- Nagata, Y., & Bräysy, O. (2009). Edge assembly-based memetic algorithm for the capacitated vehicle routing problem. *Networks: An International Journal*, 54(4), 205-215.

- Neo.lcc.uma.es-vehicle-routing-problem [Online image]. (2020). History. [http://neo.lcc.uma.es/vrp/vehicle-routing-problem/number of delivery points](http://neo.lcc.uma.es/vrp/vehicle-routing-problem/number_of_delivery_points). *Operations Research*, 12: 568-581.
- Ombuki-Berman, B. M., Runka, A., & Hanshar, F. (2007, May). Waste collection vehicle routing problem with time windows using multi-objective genetic algorithms. In *Proceedings of the Third IASTED International Conference on Computational Intelligence* (pp. 91-97). ACTA Press.
- Ong, H. L., Goh, T. N., Poh, K. L., & Lim, C. C. (1990). A computerised vehicle routing system for refuse collection. *Advances in Engineering Software* (1978), 12(2), 54-58.
- ONU-Hábitat, 2015. El 2015 registra el mayor número de días contaminados en el D.F. REPORTE NACIONAL DE MOVILIDAD URBANA EN MÉXICO 2014-2015. <http://70.35.196.242/onuhabitatmexico/Reporte-Nacional-de-Movilidad-Urbana-en-Mexico-2014-2015.pdf>
- Park, M. W., & Kim, Y. D. (1998). A systematic procedure for setting parameters in simulated annealing algorithms. *Computers & Operations Research*, 25(3), 207-217.
- Pillac, V., Gendreau, M., Guéret, C., & Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1), 1-11.
- Pradenas, L., Oportus, B., & Parada, V. (2013). Mitigation of greenhouse gas emissions in vehicle routing problems with backhauling. *Expert Systems with Applications*, 40(8), 2985-2991.
- Rabbani, M., Farrokhi-Asl, H., & Asgarian, B. (2017). Solving a bi-objective location routing problem by a NSGA-II combined with clustering approach: application in waste collection problem. *Journal of Industrial Engineering International*, 13(1), 13-27.
- Repoussis, P. P., Tarantilis, C. D., Bräysy, O., & Ioannou, G. (2010). A hybrid evolution strategy for the open vehicle routing problem. *Computers & Operations Research*, 37(3), 443-455.
- Rizzoli, A. E., Oliverio, F., Montemanni, R., & Gambardella, L. M. (2004). Ant Colony Optimisation for vehicle routing problems: from theory to applications. *Galleria Rassegna Bimestrale Di Cultura*, 9(1), 1-50.

- Rochat, Y., & Taillard, É. D. (1995). Probabilistic diversification and intensification in local search for vehicle routing. *Journal of heuristics*, 1(1), 147-167.
- Rousseau, L. M., Gendreau, M., & Pesant, G. (2002). Using constraint-based operators to solve the vehicle routing problem with time windows. *Journal of heuristics*, 8(1), 43-58.
- Salari, M., Toth, P., & Tramontani, A. (2010). An ILP improvement procedure for the open vehicle routing problem. *Computers & Operations Research*, 37(12), 2106-2120.
- Schadschneider, A., Klüpfel, H., Kretz, T., Rogsch, C., & Seyfried, A. (2009). Fundamentals of pedestrian and evacuation dynamics. In *Multi-Agent Systems for Traffic and Transportation Engineering* (pp. 124-154). IGI Global.
- Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., & Dueck, G. (2000). Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, 159(2), 139-171.
- Scilab, 2017 Optimization in SCILAB
<https://www.scilab.org/sites/default/files/OptimizationInScilab.pdf>
- SECTUR, 2019. Estudio de Competitividad Turística del Destino Cuernavaca, Morelos. Recuperado de <http://www.sectur.gob.mx/wp-content/uploads/2015/02/PDF-Cuernavaca.pdf>.
- Shaw, P. (1998, October). Using constraint programming and local search methods to solve vehicle routing problems. In *International conference on principles and practice of constraint programming* (pp. 417-431). Springer, Berlin, Heidelberg.
- Solís Ávila, J. C., & Sheinbaum Pardo, C. (2016). Consumo de energía y emisiones de CO₂ del autotransporte en México y escenarios de mitigación. *Revista internacional de contaminación ambiental*, 32(1), 7-23.
- Solomon-benchmark (2010). Solomon Benchmark. Retrieved May 02, 2020, from <https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark/100-customers/>
- Solomon, M. M. (1984). Vehicle routing and scheduling with time window constraints: Models and algorithms (No. 84-17364 UMI).
- Suthikarnnarunai, N. (2008, March). A sweep algorithm for the mix fleet vehicle routing problem. In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (Vol. 2, pp. 19-21).

- Taillard, É. D., & Gambardella, L. M. (1997). Adaptive memories for the quadratic assignment problem. Technical Report IDSIA-87-97, IDSIA, Lugano, Switzerland.
- Taillard, É., Badeau, P., Gendreau, M., Guertin, F., & Potvin, J. Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science*, 31(2), 170-186.
- Tasan, A. S., & Gen, M. (2012). A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. *Computers & Industrial Engineering*, 62(3), 755-761.
- Tech Data Corporation. (2020). Ciudades Inteligentes. Website. <https://blog.techdata.com/ts/latam/en-sus-marcas...-listos...-ces2018>
- Teixeira, J., Antunes, A. P., & de Sousa, J. P. (2004). Recyclable waste collection planning—a case study. *European Journal of Operational Research*, 158(3), 543-554.
- Thangiah, S. R., Osman, I. H., & Sun, T. (1994). Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows. Computer Science Department, Slippery Rock University, Technical Report SRU CpSc-TR-94-27, 69.
- Tillman, F. A. (1969). The multiple terminal delivery problem with probabilistic demands. *Transportation Science*, 3(3), 192-204.
- Toth, P., & Vigo, D. (1998). Exact solution of the vehicle routing problem. In *Fleet management and logistics* (pp. 1-31). Springer, Boston, MA.
- TSP (2020). The travelling salesman problem with integer programming and gurobi [Online image]. (2020). History. <http://noxexo.com/search/traveling-salesman.html>
- Tung, D. V., & Pinnoi, A. (2000). Vehicle routing–scheduling for waste collection in Hanoi. *European Journal of Operational Research*, 125(3), 449-468. University of Pennsylvania
- Vakhania, N., Chinos, E., & Zavala, C. (2015). An efficient Heuristic for a discrete optimization problem. *Journal of Computer Science*, 2(2), 39.
- Van Landeghem, H. R. G. (1988). A bi-criteria heuristic for the vehicle routing problem with time windows. *European Journal of Operational Research*, 36(2), 217-226.

Wang, C., Mu, D., Zhao, F., & Sutherland, J. W. (2015). A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows. *Computers & Industrial Engineering*, 83, 111-122.

Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary computation*, 3(2), 82-102.

Cuernavaca, Morelos a 16 de Julio del 2020.

DR. AUGUSTO RENATO PÉREZ MAYO
SECRETARIO DE INVESTIGACIÓN Y POSGRADO DE LA FCAeI
PRESENTE

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de la Maestría en Administración de Organizaciones, del estudiante Diego Gustavo Hurtado Olivares, con matrícula 10023093, con el título **Aplicación del modelo WCVRPTW al problema de recolección de residuos sólidos de empresas del sector turístico**, por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que el estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente
Por una humanidad culta
Una universidad de excelencia



Dr. José Alberto Hernández Aguilar
Profesor- investigador
Facultad de Contaduría, Administración e Informática



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

JOSE ALBERTO HERNANDEZ AGUILAR | Fecha:2020-11-10 10:27:11 | Firmante

rBoumScGm3b/OTuHfXeUQffp1ue2lGcOwZaujcOgXo5On936FjYDduDmEbZxYeLh2UXXbOYzvjpXLfDc9Oj7x0MNz8nbVRF/nEBFIhR3LThP2D6i4XtevoRie5bvMrmuJJYmpAC
YmH11WJ9aDi3UnoLi2GBQ2TdK2QhFnEZpXyXifcQetnDhnBzpqhxJOPCBfuZhgRWCd/aXI9dt5Zf6wy2pnHv6Km3D2McFL17BkgyAGH2IH86p5/XdGCJ41eWQaXeg60gBKvs8
mlPE/Aap+UBv72bxp+sGpO2/oWTsZo6x6N3FaNRVT+42LdCcL4wAj8n58KoRzh4bkNBvc4J0g==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o
escaneando el código QR ingresando la siguiente clave:



iKuCSO

<https://efirma.uaem.mx/noRepudio/lyjjqCOsTt2v589gjldJPpawVjjg79dC>





UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS



Cuernavaca, Morelos a 9 de noviembre del 2020.

DR. AUGUSTO RENATO PÉREZ MAYO
SECRETARIO DE INVESTIGACIÓN Y POSGRADO DE LA FCAeI
PRESENTE

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de la Maestría en Administración de Organizaciones, del estudiante Diego Gustavo Hurtado Olivares, con matrícula 10023093, con el título **Aplicación del modelo WCVRPTW al problema de recolección de residuos sólidos de empresas del sector turístico**, por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que el estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente
Por una humanidad culta
Una universidad de excelencia

Dr. José Crispín Zavala Díaz
Profesor- investigador
Facultad de Contaduría, Administración e Informática



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

JOSE CRISPIN ZAVALA DIAZ | Fecha:2020-11-09 21:25:41 | Firmante

RxmnyXXc8g0v2UJKAgEPjllDIV/TsqZCnEt7Xr0GkVdgojYST7bcTMqvdYRJQ7LTMnTdf/Owz3LXCgPr/fWunPeYIRVuLjOWZ9ddGAtQkgDUb4sw3xcxijdrLov4ZUaNRc16UamkUCdwpcvU6j7JppMdXKaXZ4194zldWX/zwuMvVSTqMPPahaRtelD2Xi6Fpzu8Da8165tooMVf2MJlezAKWrcINkuWw/+1mcK5m121JdVcCa56b0pQwpEXHnztP46zmYITbMZBdWZFoS5fV13ptLL306aaD2mi+Tecq4wXswmd3RsTLmkxrTVV2wAvJ0ZlmysclKTGukEuxqKpXhSQ==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o escaneando el código QR ingresando la siguiente clave:



[ko9ES2](#)

<https://efirma.uaem.mx/noRepudio/yxINXPn3UWMwFFrTmKcssC89IKQTEqZ3>





UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS



Cuernavaca, Morelos a 7 de Octubre del 2020.

DR. AUGUSTO RENATO PÉREZ MAYO
SECRETARIO DE INVESTIGACIÓN Y POSGRADO DE LA FCAeI
PRESENTE

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de la Maestría en Administración de Organizaciones, del estudiante Diego Gustavo Hurtado Olivares, con matrícula 10023093, con el título **Aplicación del modelo WCVRPTW al problema de recolección de residuos sólidos de empresas del sector turístico**, por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que el estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente
Por una humanidad culta
Una universidad de excelencia

DR. ALBERTO OCHOA
NOMBRE DEL DOCENTE
Profesor- investigador
Facultad de Contaduría, Administración e Informática



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

CARLOS ALBERTO OCHOA ORTIZ | Fecha:2020-11-17 15:14:44 | Firmante

fiT5vf773YuR1ZQwmdd+PmmMYNS52BRc0MyK4mSz2UGXnPXPtHJHkBXASL1izOfiiG5bOrmBctlUbg62cm9GI+XLapvKIVR4gW/XdNkTdKQIIOdH65hfK551YPSd2aB9FUj0J1dH9SxKDUWWGOiCh27+WCPuZ8Xh5GGBEoLoPm8xtNlivdk5HjvFNp8+8QP2kNsmIusdgeexU52R2qWJBNS9dou3gkImT5vseWMgyrud3Nr7Q1bAozqRstIwciUc+JafFYDalZGBkwd9IKwmArlaAilTuiTZhCwrD858V5o0M5KtnKeri/H5yhDVuGpTFWJuUV++iqe8dIRMT6A/w==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o escaneando el código QR ingresando la siguiente clave:



1JUcAx

<https://efirma.uaem.mx/noRepudio/mWUkvmPDT1jlkjAt6M9hNh4TVfWHgPu>





UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS



Cuernavaca, Morelos a 20 de octubre del 2020.

DR. AUGUSTO RENATO PÉREZ MAYO
SECRETARIO DE INVESTIGACIÓN Y POSGRADO DE LA FCAei
PRESENTE

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de la Maestría en Administración de Organizaciones, del estudiante Diego Gustavo Hurtado Olivares, con matrícula 10023093, con el título **Aplicación del modelo WCVRPTW al problema de recolección de residuos sólidos de empresas del sector turístico**, por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que el estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente
Por una humanidad culta
Una universidad de excelencia

Federico Alonso Pecina
Profesor- investigador
Facultad de Contaduría, Administración e Informática



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

FEDERICO ALONSO PECINA | Fecha:2020-10-20 22:16:54 | Firmante

Xy+8gEreJmbtNOeOkgt3Da4koWsTA0DLEInocawZmUNal8umLmaHG5zYZ9FfejC0NV/9EOGJ9uGypIS9sc0jEegUh4ODthe3sTylGVuRN+fAKwWOhzGLszlGcOUom9oUpaG
YHvBEIXrmvWWKWmLiAwXASq+lvXl9RZyt9vhmteN7Q0BqwuVSti+TFDc/AG8ucM81McCapHsrVZyxmr2LJsM2gvrLYU5JtJqcvUe3rfEwPHTx+5F3rap/+BQ9YEYE/i+jiNeVbzadV
oQljwniJUOuJRPSPi/SFPIIE1p1webJkYh1XSuX1xz2eCZQ3xneuJlcat0asUGa+pR9DypTA==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o
escaneando el código QR ingresando la siguiente clave:



[oUuMrG](#)

<https://efirma.uaem.mx/noRepudio/9Lehl9ZF8CsHSBVKBsyD8Mwq1pkaRDhc>



Cuernavaca, Morelos a 12 de Octubre del 2020.

DR. AUGUSTO RENATO PÉREZ MAYO
SECRETARIO DE INVESTIGACIÓN Y POSGRADO DE LA FCAei
PRESENTE

En mi carácter de revisor de Tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de la Maestría en Administración de Organizaciones, del estudiante Diego Gustavo Hurtado Olivares, con matrícula 10023093, con el título **Aplicación del modelo WCVRPTW al problema de recolección de residuos sólidos de empresas del sector turístico**, por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, concluyo que el trabajo se caracteriza por el establecimiento de objetivos académicos pertinentes y una metodología adecuada para su logro. Además construye una estructura coherente y bien documentada, por lo cual considero que los resultados obtenidos contribuyen al conocimiento del tema tratado.

Con base en los argumentos precedentes me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que el estudiante continúe con los trámites que esta Secretaría de Investigación y Posgrado tenga establecidos para obtener el grado mencionado.

Atentamente
Por una humanidad culta
Una universidad de excelencia

Dr. Martín Heriberto Cruz Rosales
Profesor- investigador
Facultad de Contaduría, Administración e Informática



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

MARTIN HERIBERTO CRUZ ROSALES | Fecha:2020-11-11 21:35:25 | Firmante

UK5ZgRM20DhpVxYYbG1mHP4/+Ztfy7ScH4CHsKFdnUwtZfn3d43QirCRW3YD9ZjaBaeel6vyuzcdsq1nyh/db60o4DQFXhoYDPFONBi2q9gyuAxwVY+47UYdVivFCoebjpx2IKLv
qaj8m9fxhvcgc6ofZQ8GluisDILa3jvHMwusCtj5x8TDBSc8iZ9WJg3y073GxEXoChYBxIRFt5QIHk2jF0kOdtJ4+XKiytdPv45v46caHo6y+qSlvpJnjPdOCRB3dF3lRnghLORfPYff84pa
jvtkXggyOXUwC10C9b8tUr0cH++onB2grUPjjasNFqOGIG+nRDvbAKdDrzQ==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o
escaneando el código QR ingresando la siguiente clave:



[uEp6sN](#)

<https://efirma.uaem.mx/noRepudio/TmcRqB8NRmVXiLOGtRuR1JYNb8h3SIAf>

