

**SOLUCION DEL PROBLEMA DE RUTEO DE VEHICULOS DEPENDIENTES DEL TIEMPO
UTILIZANDO UN ALGORITMO GENETICO MODIFICADO**

FREDY ALEXANDER GUASMAYAN GUASMAYAN

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍA INDUSTRIAL
MAESTRÍA EN INVESTIGACIÓN OPERATIVA Y ESTADÍSTICA
PEREIRA - RISARALDA
FEBRERO DE 2014**

**SOLUCION DEL PROBLEMA DE RUTEO DE VEHICULOS DEPENDIENTES DEL TIEMPO
UTILIZANDO UN ALGORITMO GENETICO MODIFICADO**

FREDY ALEXANDER GUASMAYAN GUASMAYAN

**Tesis presentada como requisito para optar al título de Magíster en Investigación
Operativa y Estadística**

**Directora
M.Sc. ELIANA MIRLEDY TORO OCAMPO**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍA INDUSTRIAL
MAESTRÍA EN INVESTIGACIÓN OPERATIVA Y ESTADÍSTICA
PEREIRA - RISARALDA
FEBRERO DE 2014**

RESUMEN

El ruteo de vehículos corresponde a un tipo de problemas combinatoriales que en la medida en que se pretenda obtener la o las rutas óptimas que permitan atender completamente un conjunto de clientes determinado, el problema se vuelve muy complicado computacionalmente, de forma que para un número de clientes mayor a 20 con una sola ruta se tiene $20!/2$ combinaciones posibles como se indica en [1], es decir 12×10^{17} combinaciones, siendo esto casi imposible de resolver computacionalmente con los métodos exactos existentes, los cuales resuelven el problema por medio del modelado matemático, es decir buscando cada una de las combinaciones posibles y teniendo en cuenta cada una de las restricciones del problema.

Las restricciones en los modelos matemáticos enmarcan aspectos tales como la capacidad de los vehículos, el visitar a un cliente a la vez, la satisfacción de la demanda que requiere cada cliente, entre otros aspectos propios del problema de acuerdo con la aplicación.

En el caso de problemas de ruteamiento reales y que contengan instancias de gran tamaño, existen diferentes métodos de solución como son los algoritmos heurísticos y metaheurísticos, los cuales por medio de iteraciones lógicas permiten obtener soluciones óptimas o cercanas al óptimo en determinados casos. Por otra parte cuando el problema es aplicado a la realidad genera otro tipo de restricciones adicionales como puede ser el tiempo en que se debe entregar un producto, la capacidad de carga de los vehículos, el número de vehículos, factores ambientales, factores de tráfico, entre muchos otros. Tal es el caso del ruteo de vehículos con dependencia del tiempo, en el que el tiempo de salida del vehículo determina el costo de la ruta, puesto que según la hora en que se realice el recorrido, se tendrá mayor o menor tráfico, lo que conlleva a elevar el costo no solo por distancia recorrida, sino por el tiempo que se gasta el vehículo en recorrer la ruta.

Para esto se propone resolver el problema de ruteo de vehículos con dependencia del tiempo (TDCVRP) como una variante del ruteo de vehículos con capacidad fija (CVRP), el cual considera el tiempo u horario en el que se hace el recorrido de la ruta de los vehículos para hacer la entrega de productos a cada cliente.

La restricción adicional del tiempo es de gran importancia en los problemas de ruteo reales, puesto que los costos en el recorrido de la o las rutas de los vehículos son elevados debido al tráfico y la congestión en determinadas horas del día, lo que genera gastos de operación reflejados en combustible y tiempo, además de considerar a la congestión vehicular como fuente de estrés a los conductores y foco de contaminación ambiental.

Por otra parte este tipo de restricciones permiten cumplir de manera más efectiva las solicitudes del cliente, puesto que el tomar una ruta considerando únicamente la distancia para llegar desde un punto a otro no garantiza el llegar a tiempo a las entregas del

producto a los clientes, por esto el considerar el horario de cubrimiento de las rutas permite mejorar y eliminar posibles retrasos dando mayor confiabilidad a las empresas.

El problema del TDVRP se aborda por medio de técnicas heurísticas como herramientas de inicio para la metaheurística del algoritmo genético modificado propuesto por Chu-Beasley, los cuales toman soluciones dadas por métodos heurísticos como la técnica de ahorros, técnicas de barrido, intercambios de nodos y aristas, entre otras, y asignarlas a una población inicial con el fin de encontrar en la mezcla de estas soluciones de forma aleatoria otras soluciones que mejoren la función objetivo determinada por el costo de las rutas en los vehículos.

El algoritmo se prueba con diferentes instancias del problema de CVRP conocidas mostrado en el capítulo 5 como resultados de la ejecución del algoritmo genético para instancias publicadas en la literatura especializada, para luego implementar dicho algoritmo en el caso particular de la distribución de productos lácteos de la empresa COLÁCTEOS en la ciudad de San Juan de Pasto, Nariño. Para lo cual se realiza una georeferenciación de los clientes de la ciudad a los que distribuye la empresa y se obtiene la matriz de costos dada por las distancias entre puntos o clientes, adicionalmente se considera el horario en que se realiza el recorrido de los vehículos y la capacidad máxima que estos tienen junto con la demanda de las tiendas en la ciudad. Con base en la matriz de costos se genera el mejoramiento de las rutas haciendo uso del algoritmo genético modificado. Para el cual se obtienen resultados que disminuyen los costos en la empresa en más del 34%.

TABLA DE CONTENIDO

1.	INTRODUCCIÓN.....	12
1.1	PLANTEAMIENTO DEL PROBLEMA.....	14
1.2	DELIMITACIÓN	16
1.3	OBJETIVOS.....	16
1.1.1	Objetivo general	16
1.1.2	Objetivos específicos.....	16
1.4	JUSTIFICACIÓN	17
1.5	DISEÑO METODOLÓGICO.....	17
	FASE I: Diagnostico del Problema	18
	FASE II: Identificación de un Modelo Matemático	18
	FASE III: Obtención de una Solución a partir del Modelo	18
	FASE IV: Prueba del Modelo de solución.	19
	FASE V: Establecimiento de Controles sobre la Solución.....	19
	FASE VI: Validación de la Metodología de Solución.....	19
2.	PROBLEMA DE RUTEAMIENTO CON DEPENDENCIA DEL TIEMPO TDVRP	20
2.1	ANTECEDENTES.....	20
2.2	MARCO REFERENCIAL	25
2.3	MARCO CONCEPTUAL	29
2.3.1	Problemas característicos en la distribución y transporte.	29
2.3.2	Modelo matemático del problema de ruteo de vehículos con dependencia del tiempo	34

2.3.3	Métodos de solución a problemas combinatoriales.....	40
3.	METODOLOGÍA DE SOLUCION PARA EL PROBLEMA TDVRP.....	51
3.1	ALGORITMO GENÉTICO MODIFICADO EN TDVRP.....	51
3.1.1	Representación del cromosoma:.....	53
3.1.2	Ajuste de parámetros:.....	56
3.1.3	Selección:	59
3.1.4	Cruce de individuos:	60
3.1.5	Mutación:	61
3.1.6	Modificación de la población:	62
3.1.7	Etapa de mejoramiento de las soluciones:	62
3.2	IMPLEMENTACIÓN DE HEURÍSTICAS Y METAHEURÍSTICAS PARA LA CONFIGURACIÓN DE LA POBLACIÓN INICIAL DEL ALGORITMO GENÉTICO.....	63
3.2.1	Heurística de intercambio 2opt.....	64
3.2.2	Heurística de inserción del más barato.	65
3.2.3	Heurística de la ruta mínima o vecino más cercano.	69
3.2.4	Heurística de ahorros de Clarke y Wriht.....	72
3.2.5	Técnica del barrido.....	77
3.2.6	Metaheurística de Colonia de hormigas.	78
3.2.7	Metaheurística Colonia de hormigas con intensificación.....	78
3.2.8	Intensificación algoritmo genético usando Metaheurística VNS.....	79
4.	CARACTERIZACION DE RUTAS, MATRIZ DE COSTOS Y DEMANDAS PARA RUTAS DE LA EMPRESA COLÁCTEOS EN LA CIUDAD DE SAN JUAN DE PASTO (NARIÑO).	81
4.1	DESCRIPCIÓN DE LA EMPRESA COLACTEOS.....	81

4.2	RUTAS INICIALES POR VEHICULOS DE LA EMPRESA COLÁCTEOS EN LA CIUDAD DE SAN JUAN DE PASTO.....	83
4.3	ASIGNACION DE DEMANDA A CADA SECTOR.....	87
4.4	CALCULO DE MATRIZ DE DISTANCIAS ENTRE SECTORES PARA CARACTERIZAR COSTOS.....	88
5	RESTRICCION DE DEPENDENCIA DEL TIEMPO PARA LAS RUTAS	90
5.1	TOMA DE DATOS Y CALCULO DE DENSIDAD VEHICULAR	90
5.2	ANALISIS ESTADÍSTICO.....	90
6	PRUEBAS Y RESULTADOS OBTENIDOS	93
7	CONCLUSIONES Y RECOMENDACIONES.....	106
8	REFERENCIAS BIBLIOGRÁFICAS	109
9	ANEXOS	115
9.1	PROGRAMA PRINCIPAL PARA ALGORITMO GENÉTICO.....	115
9.2	ANÁLISIS DE ARTICULOS RELACIONADOS.....	125
9.3	FUNCIONES PARA ASIGNACIÓN DE DEMANDAS A LOS DIFERENTES SECTORES	142
9.4	TABLAS DE RESULTADOS DE ANÁLISIS DE VARIANZA PARA LA DENSIDAD VEHICULAR.....	143
9.5	PROGRAMACIÓN DE SERVICIOS DE GOOGLMAPS PARA OBTENCIÓN DE MATRIZ DE DISTANCIAS	160
9.6	TABLAS DE DATOS PARA CARACTERIZACIÓN DE LA RESTRICCIÓN DE DEPENDENCIA DEL TIEMPO.....	165
9.7	TABLAS DE DENSIDAD VEHICULAR.....	173
9.8	TABLAS DE COORDENADAS EN LATITUD Y LONGITUD PARA LAS RUTAS DE LOS 24 VEHÍCULOS DE COLÁCTEOS POR SECTORES	179
9.9	CALCULO DE SOBRECOSTO DE LAS RUTAS DADAS LAS CONDICIONES TEMPORALES....	180

LISTA DE TABLAS

<i>Tabla 2-1. Revisión del estado del arte.</i>	26
<i>Tabla 3-1 Matriz de distancias para 10 ciudades.</i>	55
<i>Tabla 3-2. Pruebas para mutación de un gen y un punto de recombinación</i>	57
<i>Tabla 3-3. Pruebas para mutación de dos genes y un punto de recombinación</i>	57
<i>Tabla 3-4. Pruebas para mutación de un gen y dos puntos de recombinación.</i>	58
<i>Tabla 3-5. Pruebas para mutación de dos genes y dos puntos de recombinación</i>	58
<i>Tabla 3-6. Matriz de distancias para 5 ciudades</i>	67
<i>Tabla 3-7. Matriz de distancias para 5 ciudades</i>	71
<i>Tabla 3-8. Matriz de distancias para 5 ciudades en técnica de ahorros</i>	74
<i>Tabla 4-1. Ruta del vehículo 1 de Colácteos por sectores y coordenadas de latitud y longitud</i>	85
<i>Tabla 4-2. Resultados de los sectores correspondientes a cada vehículo</i>	87
<i>Tabla 4-3. Demanda de canastos por tiendas y por sectores de cada vehículo.</i>	87
<i>Tabla 5-1. Combinación del diseño factorial</i>	90
<i>Tabla 6-1. Resultados para las corridas del algoritmo genético para diferentes instancias.</i>	93
<i>Tabla 6-2. Valor de la función objetivo para instancia de Augerat60k9</i>	94
<i>Tabla 6-3. Rutas de la solución a la instancia de Augerat60k9</i>	94
<i>Tabla 6-4. Secuencia de visitas a las ciudades por cada vehículo.</i>	95
<i>Tabla 6-5. Respuestas de la función objetivo para 20 corridas del algoritmo genético en 153 sectores de la ciudad de San Juan de Pasto.</i>	95
<i>Tabla 6-6. Rutas para sectores en la ciudad de San Juan de Pasto.</i>	97
<i>Tabla 6-7. Costos en distancia y combustible para rutas con las que actualmente cuenta la empresa.</i>	97
<i>Tabla 6-8. Resumen de distancia y tiempo de recorrido de las 18 rutas para distribución de productos lácteos en la ciudad de San Juan de Pasto.</i>	99
<i>Tabla 6-9. Comparativo de rutas de Colácteos con Rutas propuestas</i>	100
<i>Tabla 6-10. Velocidad media y crítica para calles clasificadas en categoría alta.</i>	101
<i>Tabla 6-11. Velocidad media y crítica para calles clasificadas en categoría media.</i>	101
<i>Tabla 6-12. Velocidad media y crítica para calles clasificadas en categoría baja.</i>	102
<i>Tabla 6-13. Valores de velocidad media por cada categoría.</i>	103
<i>Tabla 6-14. Relación de velocidad media con velocidad crítica por categoría.</i>	103
<i>Tabla 6-15. Costos de ruta 1, incluyendo la congestión vehicular en horas pico</i>	104
<i>Tabla 6-16. Costo de cada ruta y total afectado por sobrecosto de tráfico en hora pico</i>	104
<i>Tabla 9-1. Resultado prueba de Levene para la variable dependiente densidad</i>	144
<i>Tabla 9-2. ANOVA para los factores de día, hora y categoría en la variable densidad vehicular</i>	144
<i>Tabla 9-3. Comparaciones múltiples para la variable Día</i>	145
<i>Tabla 9-4. Comparaciones múltiples para la variable categoría</i>	145
<i>Tabla 9-5. Pruebas de normalidad</i>	148
<i>Tabla 9-6. ANOVA Pruebas de los efectos inter-sujetos para rangos de la variable densidad vehicular</i>	149
<i>Tabla 9-7. Comparaciones múltiples para el Factor Día</i>	150
<i>Tabla 9-8. Comparaciones múltiples para el Factor Hora</i>	151

<i>Tabla 9-9. Comparaciones múltiples para el Factor Categoría de calle</i>	151
<i>Tabla 9-10. Pruebas de normalidad para la variable densidad con rangos múltiples</i>	152
<i>Tabla 9-11. Prueba de Levene para la variable dependiente densidad por rangos</i>	152
<i>Tabla 9-12. Contraste de Levene sobre la igualdad de las varianzas para la variable densidad vehicular</i>	153
<i>Tabla 9-13. Pruebas de normalidad variable densidad</i>	153
<i>Tabla 9-14. Pruebas de los efectos inter-sujetos</i>	154
<i>Tabla 9-15. Densidad vehicular para las calles de referencia</i>	156
<i>Tabla 9-16. Calles con 8 muestras de densidad en horas pico.</i>	157
<i>Tabla 9-17. Calles con menos de 8 muestras de densidad vehicular en horas pico</i>	157
<i>Tabla 9-18. Clasificación de calles según prueba estadística T para diferencia de medias de densidad vehicular</i>	159
<i>Tabla 9-19. Resumen información de rutas de vehículos según entrevista y aplicación de google maps</i>	165
<i>Tabla 9-20. Coordenadas con puntos de todos los vehículos:</i>	165
<i>Tabla 9-21. Rutas que atienden a cada sector y demanda de cada sector</i>	169
<i>Tabla 9-22. Densidad vehicular calle principal Lorenzo día sábado.</i>	173
<i>Tabla 9-23. Densidad vehicular calle principal Lorenzo día domingo</i>	173
<i>Tabla 9-24. Densidad vehicular calle principal Lorenzo día jueves</i>	174
<i>Tabla 9-25. Densidad vehicular calle 15 día sábado</i>	174
<i>Tabla 9-26. Densidad vehicular calle 15 día domingo</i>	175
<i>Tabla 9-27. Densidad vehicular calle 15 día jueves</i>	175
<i>Tabla 9-28. Densidad vehicular Capusigra día Sábado</i>	175
<i>Tabla 9-29. Densidad vehicular Capusigra día domingo</i>	176
<i>Tabla 9-30. Densidad vehicular Capusigra día jueves</i>	176
<i>Tabla 9-31. Variables categóricas y densidad vehicular</i>	177
<i>Tabla 9-32. Coordenadas en latitud y longitud para las 6 primeras rutas</i>	179
<i>Tabla 9-33. Coordenadas en latitud y longitud para las rutas de la 7 a la 12.</i>	179
<i>Tabla 9-34. Coordenadas en latitud y longitud para las rutas de la 12 a la 18.</i>	180
<i>Tabla 9-35. Asignación de sobre costos a las rutas</i>	182
<i>Tabla 9-36. Costos de cada ruta incluyendo la congestión vehicular en horas pico</i>	184
<i>Tabla 9-37. Costo de cada ruta y total afectado por sobre costo de tráfico en hora pico</i>	187

LISTA DE FIGURAS

<i>Figura 1-1 Aspecto de una solución al problema de rutas VRP.</i>	13
<i>Figura 2-1. Esquema problema m-TSP</i>	31
<i>Figura 2-2. Esquema del problema de ruteamiento de vehículos VRP.</i>	32
<i>Figura 3-1 Diagrama de flujo algoritmo genético</i>	53
<i>Figura 3-2. Vector de secuencia de clientes</i>	54
<i>Figura 3-3. Vector de vehículos del individuo.</i>	54
<i>Figura 3-4. Demanda de clientes</i>	54
<i>Figura 3-5. Vector Clientes</i>	55
<i>Figura 3-6. Demanda en orden de los clientes</i>	55
<i>Figura 3-7. Vector clientes y Vector de demandas correspondientes a cada cliente</i>	56
<i>Figura 3-8. Vector de vehículos del individuo.</i>	56
<i>Figura 3-9. Rendimiento del algoritmo para población de 30 individuos</i>	59
<i>Figura 3-10. Rendimiento del algoritmo para población de 200 individuos</i>	59
<i>Figura 3-11. Individuo de la población</i>	60
<i>Figura 3-12. Vector Padre 1</i>	60
<i>Figura 3-13. Vector Padre 2</i>	60
<i>Figura 3-14. Vectores Hijos por recombinación</i>	61
<i>Figura 3-15. Vectores Hijos</i>	61
<i>Figura 3-16. Comportamiento del porcentaje de mutación en el algoritmo genético.</i>	62
<i>Figura 3-17. Rutas solución VRP instancia de Augerat para 80 ciudades.</i>	63
<i>Figura 3-18. Diagrama de flujo para heurística 2 OPT</i>	64
<i>Figura 3-19. Vector permutado</i>	65
<i>Figura 3-20. Vector Vehículos</i>	65
<i>Figura 3-21. Diagrama de flujo para inserción del más barato</i>	66
<i>Figura 3-22. Vector demanda</i>	67
<i>Figura 3-23. Vector secuencia de clientes</i>	67
<i>Figura 3-24. Vector verificación de insertados</i>	68
<i>Figura 3-25. Vector verificación de insertados actualizado</i>	68
<i>Figura 3-26. Vector de rutas</i>	68
<i>Figura 3-27. Vector vehículos</i>	68
<i>Figura 3-28. Diagrama de flujo del vecino más cercano</i>	70
<i>Figura 3-29. Vector demanda</i>	71
<i>Figura 3-30. Vector verificación visitas</i>	71
<i>Figura 3-31. Orden de visitas</i>	72
<i>Figura 3-32. Vector rutas</i>	72
<i>Figura 3-33. Vector vehículos</i>	72
<i>Figura 3-34. Grafo para Técnica de ahorros.</i>	73
<i>Figura 3-35. Diagrama de flujo técnica de ahorros</i>	74

<i>Figura 3-36. Vector demanda</i>	74
<i>Figura 3-37. . Asignación de ruta para cada nodo</i>	75
<i>Figura 3-38. Introducción de nodo de máximo ahorro a la ruta</i>	76
<i>Figura 3-39. Vector ruta</i>	76
<i>Figura 3-40. Vector vehículos</i>	76
<i>Figura 3-41. Diagrama de flujo para técnica del barrido</i>	77
<i>Figura 3-42. Vector ruta</i>	79
<i>Figura 3-43. Combinaciones VNS</i>	79
<i>Figura 4-1 Forma de carga de productos en vehículos de la empresa Colácteos.</i>	84
<i>Figura 4-2. Ruta del vehículo 1 obtenido en la aplicación de googlemaps</i>	86
<i>Figura 6-1. Respuesta para las rutas de la instancia Augerat60k9</i>	94
<i>Figura 6-2. Ruta del vehículo 3.</i>	99
<i>Figura 9-1. Rutas solución VRP instancia de Augerat para 80 ciudades.</i>	123
<i>Figura 9-2 Secuencia de clientes en solución VRP para instancia de 80 ciudades.</i>	124
<i>Figura 9-3. Generación de código para el sector en cada punto de la ruta.</i>	142
<i>Figura 9-4. Función para encontrar las rutas que aportan a cada sector.</i>	143
<i>Figura 9-5. Función para el cálculo de demanda de cada sector.</i>	143
<i>Figura 9-6. Promedio de densidad para las variables Día Vs Hora</i>	146
<i>Figura 9-7. Promedio de densidad para las variables Día Vs Categoría de calle</i>	147
<i>Figura 9-8. Promedio de densidad para las variables Hora Vs Categoría de calle</i>	147
<i>Figura 9-9. Prueba de independencia variable densidad vehicular</i>	149
<i>Figura 9-10. Prueba de independencia para la variable densidad vehicular con rangos múltiples</i>	152
<i>Figura 9-11. Supuesto de independencia variable densidad</i>	154
<i>Figura 9-12 Resultado de ANOVA usando prueba no paramétrica de Kruskal Wallis</i>	155
<i>Figura 9-13. Codificación prueba T para diferencia de medias con datos desbalanceados</i>	155
<i>Figura 9-14 Función de inicialización de API de google.</i>	161
<i>Figura 9-15. Variables de inicio y destino de los sectores a visitar.</i>	161
<i>Figura 9-16 Función Recorrido. Solicitud de servicio de googlemaps para realizar el recorrido de un punto origen hacia un destino en coordenadas de latitud y longitud</i>	162
<i>Figura 9-17 Función calcular ruta.</i>	163
<i>Figura 9-18 Función calcular distancia.</i>	163
<i>Figura 9-19. Aplicativo para calcular distancia entre puntos.</i>	164
<i>Figura 9-20. Matriz de distancias de 154x154 puntos.</i>	164
<i>Figura 9-21. Ruta por calles de la ciudad de San Juan de Pasto para el vehículo 2 de Colácteos.</i>	181

1. INTRODUCCIÓN

Actualmente en todos los campos productivos se pretende el mejoramiento continuo de cada uno de sus procesos, por consiguiente es importante la realización de diferentes estudios sobre optimización de procesos que permitan llevar a cabo soluciones que resuelvan de manera efectiva un problema, sobre todo en actividades productivas propias del sector empresarial a nivel regional y nacional, puesto que el impacto que genera está relacionado directamente con el costo de los productos terminados en la producción de las cadenas de suministro y es de vital importancia encontrar la solución óptima que minimice dichos costos mejorando las condiciones del producto terminado y a su vez que permita a la empresa del sector tanto regional como nacional tener un grado de competitividad adecuado en mercados internacionales y poder continuar con los procesos generando oportunidades de empleo y solvencia económica para la región y el país. Para ello existen herramientas como la optimización que permite mediante la modelación matemática encontrar el valor mínimo o máximo del objetivo que pretenda el modelo, sin embargo cuando los problemas presentan un número grande de combinaciones posibles los métodos clásicos de solución con programación lineal o no lineal presentan complejidad computacional de tal manera que es casi imposible encontrar la solución óptima puesto que la búsqueda de un valor mínimo o máximo en la función matemática correspondiente al modelo junto con sus restricciones presenta tiempos computacionales prohibitivos o las posibilidades son casi que infinitas y el tiempo que tardaría un procesador electrónico sería demasiado alto y más cuando los problemas deben tener soluciones con valores enteros, para los cuales existen métodos de solución exactos que presentarían el mismo problema de soluciones en un tiempo computacional muy alto, para ello se plantean algoritmos que a pesar de no encontrar soluciones totalmente óptimas, encuentran soluciones muy cercanas al valor mínimo o máximo del problema y en tiempos computacionales relativamente bajos, a estos métodos se los denomina algoritmos de construcción o heurísticas los cuales reducen el tiempo computacional de forma significativa en los problemas combinatoriales, sin embargo cuando los problemas crecen, incluso dichas heurísticas generan dificultad en la búsqueda de las soluciones, para ello se procede a buscar algoritmos que permitan encontrar aleatoriamente una solución, a estos algoritmos se los conoce como metaheurísticas.

En el caso de algunos problemas de optimización combinatoria, las variables de decisión pueden ser de tipo entero, como es el caso de la selección de una ruta para visitar a diferentes ciudades, cargar un número definido de productos, usar diferentes tipos de vehículos, entre otros, además dichas variables por otra parte en problemas relacionados con el transporte como lo es el agente viajero, ruteo de vehículos y sus derivados, entre

otros, en general tienen valores de tipo entero y binario en los modelos matemáticos que los representan, es decir que las variables definen si se toma o no una ruta, o si un vehículo visita a un cliente o no, etc. Esto hace que la solución de dichos problemas sea de mayor complejidad aumentando las iteraciones de los algoritmos de solución y por ende aumentando el tiempo computacional para encontrar valores óptimos en el espacio de soluciones que tenga dicho modelo matemático en este tipo de problemas denominados de programación entera.

El problema del ruteo de vehículos dependientes de tiempo se considera una variante del llamado VRP (*Vehicle routing problem*) o problema de ruteamiento de vehículos en el que se toma en cuenta el horario en que se dan los viajes de los vehículos, lo cual hace que se restrinja el problema a las condiciones de las vías en determinadas horas del día, o algunos días de la semana, etc., permitiendo que las soluciones cambien en función de la probabilidad de que una vía de determinada ruta se encuentre más o menos congestionada.

Este problema se fundamenta principalmente en la generalización del TSP o problema del agente viajero (*Traveling salesman problem*) con múltiples agentes viajeros, donde existe una demanda asociada a cada ciudad y una capacidad determinada de transporte para cada uno de los vehículos. Aquí, el objetivo puede ser reducir al mínimo posible la suma de la distancia recorrida por todas las rutas, el número de vehículos, o una combinación de ambos criterios.

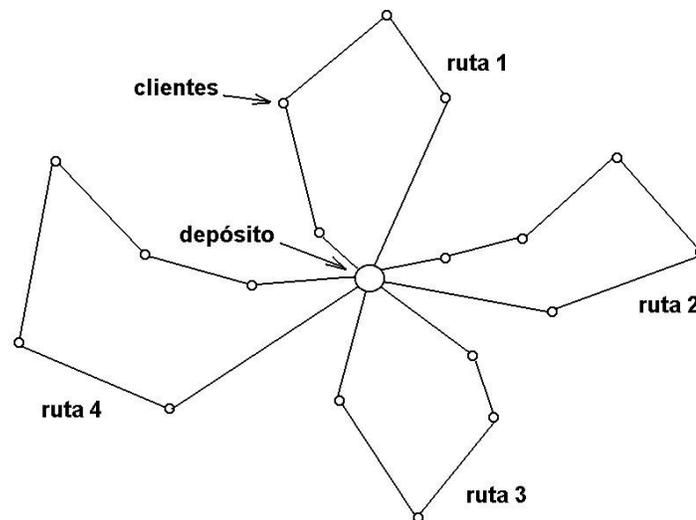


Figura 1-1 Aspecto de una solución al problema de rutas VRP.

La mayor parte de los problemas de ruteo de vehículos son generalizaciones del TSP. En ese sentido, puede considerarse el VRP más simple. El tiempo de cálculo necesario para resolver el TSP se incrementa con rapidez a medida que aumenta el número de ciudades n . En un caso general el número de rutas factibles que debe considerarse es:

$$\text{Combinaciones} = \frac{(n-1)!}{2} \quad (1-1)$$

Puesto que hay $(n - 1)$ posibilidades para la primera ciudad después de la ciudad de residencia del agente, $(n - 2)$ posibilidades para la siguiente ciudad y así sucesivamente. El denominador 2 surge porque cada ruta presenta una ruta inversa equivalente con la misma distancia (TSP simétrico). El VRP es un problema en los que se debe determinar una serie de rutas para una flota de vehículos basados en uno o más depósitos, para un cierto número de ciudades o clientes geográficamente dispersos. Es uno de los problemas de optimización combinatorial y programación lineal entera más desafiante que existe por su complejidad y por pertenecer al tipo *NP-Hard* o *NP difícil* según lo define Rinooy Kan en 1981.

La sigla NP indica que un problema no presenta una solución algorítmica en tiempo polinomial, así un problema A pertenece a la categoría *NP-difícil* si para todo problema B de *NP* existe un algoritmo que funciona en tiempo polinomial y es capaz de reinterpretar toda instancia del problema A en una instancia del problema B . indicado en [2].

De este modo debido a la gran explosión combinatorial de los problemas que pertenecen a *NP Hard* se precisa el uso de algoritmos aleatorios que resuelvan este problema con valores cercanos al óptimo pero en tiempos computacionales cortos.

En este proyecto en particular se pretende aplicar herramientas heurísticas como soluciones de inicio en la búsqueda de soluciones mejoradas del problema **TDVRP** usando vehículos de capacidad fija y homogénea.

1.1 PLANTEAMIENTO DEL PROBLEMA

La optimización significa encontrar el valor máximo o mínimo de una función matemática planteada en un modelo que describa y restrinja el problema, definida en un dominio en el que de acuerdo al problema que en la realidad de las empresas e industrias se desea elevar los rendimientos económicos y la efectividad de sus procesos con los recursos mínimos ya sean de personal, presupuestos, maquinaria, equipos y en muchos casos el tiempo, o simplemente requisitos mínimos de producción que se tienen que cumplir de alguna manera; esto condiciona la elección de la solución adecuada, ya sea a nivel estratégico, táctico e incluso operativo. Por lo tanto, el propósito perseguido al tomar una decisión consiste en llevar a cabo un plan que le permita a las empresas solucionar sus

problemas de una manera óptima o efectiva que genere respuestas al objetivo de manera que el beneficio sea máximo con los mínimos costos.

Normalmente al plantear una solución a un problema que pretenda minimizar costos y obtener el máximo beneficio, se debe iniciar por recolectar la información pertinente que permita describir matemáticamente dicho problema en un modelo, estableciendo así las características que permiten visualizar un espacio de soluciones, los diferentes métodos para su solución y el alcance computacional; esto con el fin de encontrar un valor óptimo o adecuado a la respuesta esperada para el problema; sin embargo en ocasiones la complejidad de las condiciones reales para plantear un modelo preciso conlleva a incrementar la dificultad para implementar algoritmos de solución exacta, por ello en estos problemas combinatoriales que en su medida de dificultad para el procesamiento computacional, los métodos de solución pueden ser inviables debido a la gran cantidad de procesos que se deban realizar para llegar a la solución óptima exacta en un tiempo adecuado, de modo que los problemas de optimización planteados frecuentemente se deben resolver con métodos aproximados que proporcionan soluciones factibles y satisfactorias.

Un tipo concreto de problemas de optimización son los denominados problemas de optimización combinatoria. En los que sus variables de decisión son enteras, es decir que sus soluciones están formadas por ordenaciones o subconjuntos de números naturales. Estos problemas son difíciles de resolver debido a la forma de crecimiento exponencial de las combinaciones de soluciones en relación con el número de clientes. De hecho, en la literatura se mencionan soluciones al problema de enrutamiento de vehículos de más o menos cien clientes como lo referencia V. Yepes en su tesis [1]. Por lo que es importante plantear soluciones viables con una respuesta de costos cercanos al valor óptimo por medio de algoritmos de aproximación que proporcionen dichas soluciones en un tiempo razonable.

Por otra parte los problemas de ruteamiento de vehículos en la realidad presenta un gran número de condiciones que afectan al modelo que lo describe matemáticamente, haciéndolo más difícil de resolver de manera exacta, estas condiciones son por ejemplo la capacidad del vehículo, la cual puede ser homogénea o no; otra condición puede ser el número de depósitos donde se recoge la carga, otra es la posibilidad de que el cliente reciba y devuelva carga, incluso la forma de los productos que se lleva en los vehículos puede afectar a la capacidad de estos puesto que se puede definir dicha capacidad en volumen, peso, o si es delicada o no, etc.

Otro factor importante es el tiempo en el que se debe entregar los productos a los clientes. De modo que con cada condición que se tenga en una situación real, el problema tendría mayor dificultad en resolverse, por esto en este proyecto se pretende tomar el

problema con un valor de capacidad de los vehículos homogéneo y con la adición de la restricción del tiempo que genera un costo adicional si el vehículo toma una ruta en cierta hora del día para la cual se tenga mayor congestión vehicular.

Este problema es el más adecuado en la aproximación a la realidad de las empresas lácteas de la ciudad de San Juan de Pasto a las que se pretende desarrollar una metodología de solución.

1.2 DELIMITACIÓN

En este trabajo se pretende resolver el problema de TDVRP aplicados a la empresa de distribución de productos lácteos COLÁCTEOS en la ciudad de Pasto, Nariño, la cual tiene aproximadamente 2500 clientes, por lo que se hace necesario el uso de metaheurísticas que den una solución cercana al valor óptimo.

Para esto se identifica inicialmente la matriz de costos del problema con los valores de distancia entre clientes y posteriormente se codifica el problema con algoritmos genéticos para encontrar una solución que mejore las condiciones actuales de distribución de la empresa.

1.3 OBJETIVOS

1.1.1 Objetivo general

Desarrollar e implementar una metodología basada en el Algoritmo Genético modificado que permita establecer las rutas de los vehículos en el problema de ruteo de vehículos dependientes de tiempo, teniendo en cuenta como función objetivo la minimización de los costos requeridos en la ejecución de todas las visitas desde el depósito.

1.1.2 Objetivos específicos

- Realizar un estudio del estado del arte de los diferentes problemas, modelamiento y técnicas de solución empleadas para el ruteamiento de vehículos y en particular el ruteamiento de vehículos con dependencia de tiempo.
- Definir una alternativa de codificación para el problema de ruteamiento de vehículos dependientes de tiempo que permita cuantificar el valor de la función objetivo y la adaptación de los operadores de la técnica Algoritmo Genético modificado.
- Implementar la metodología desarrollada en el lenguaje de programación de Matlab para dar solución al problema del ruteamiento de vehículos dependientes de tiempo.

- Calibrar los parámetros propios del Algoritmo Genético modificado con el fin de que el proceso de búsqueda sea más eficiente.
- Validar la metodología desarrollada a partir de la evaluación de sistemas de prueba descritos en la literatura especializada.

1.4 JUSTIFICACIÓN

Con este proyecto se pretende resolver el problema TDVRP haciendo uso de varias heurísticas que permitan generar soluciones iniciales para el mejoramiento en búsqueda de una solución más eficiente por medio de una metaheurística denominada algoritmo genético modificado de Chu-Beasley descrito en [3], el cual realizará una búsqueda iterativa de una mejor solución y en un tiempo computacional adecuado, es decir que se obtenga soluciones con errores mínimos comparados con los valores exactos encontrados en la literatura especializada y en tiempos de procesamiento eficientes comparados con la calidad de la respuesta y el tiempo que tarda el computador para dar dicha respuesta.

El interés de este proyecto tanto para la academia como para el sector productivo es de gran importancia; visto desde la academia se pretende apoyar al grupo de investigación en aplicaciones de técnicas de optimización y procesos estocásticos GAOPE de la Universidad Tecnológica de Pereira y al grupo de investigación GISMAR (Grupo de investigación ingeniería de sistemas Universidad Mariana), en los cuales se generaran diferentes alternativas de soluciones a problemas de enrutamiento de vehículos con métodos heurísticos y metaheurísticos.

En el sector productivo se busca generar apoyo a la solución de este tipo de problemas en empresas del sector lácteo de Nariño, las cuales tienen interés en mejorar la distribución de sus productos en las distintas tiendas de los barrios de la ciudad de San Juan de Pasto, las cuales requieren que los productos estén en horarios definidos, además de contar con una metodología establecida por la empresa para la carga de sus productos a los diferentes vehículos y la disposición física de estos en el depósito. Por otra parte las condiciones de la ciudad generan restricciones al problema de forma específica con el manejo del pico y placa, la congestión vehicular en horarios específicos y el deterioro de algunas vías.

Finalmente los resultados que se obtengan por medio de la solución con técnicas heurísticas y metaheurísticas se colocan a disposición de la comunidad académica y se podrá generar pruebas en el sector productivo.

1.5 DISEÑO METODOLÓGICO

Para llevar a cabo el objeto de estudio como etapa inicial del proceso se hace necesario estudiar los diferentes tipos de problemas de *ruteo de vehículos* y las técnicas de solución respectivas, según la revisión del estado del arte, posteriormente se formula y se implementa la metodología de solución y finalmente se validan los resultados obtenidos con casos de prueba de la literatura especializada. Para llevar a cabo estas etapas es de vital importancia determinar el método más adecuado para la investigación, que permita dar solución al problema.

Para el objeto de este trabajo, el método de investigación apropiado a seguir es el experimental, ya que con el desarrollo de la metodología de solución para el problema del *ruteo de vehículos con dependencia de tiempo* se pretende encontrar una solución de calidad, efectiva y sobre todo óptima considerando la minimización del tiempo de procesamiento en los algoritmos de solución y la búsqueda de rutas que minimicen el costo de los viajes de acuerdo a la distancia recorrida por los vehículos, su capacidad y el estado de las vías conforme al horario del recorrido. Para lograr este objetivo se deben realizar varias pruebas y validar la solución encontrada con el propósito de conocer si la metodología desarrollada es adecuada y presenta resultados de buena calidad.

El desarrollo metodológico está dividido en seis fases, las cuales describen el desarrollo del proyecto por medio de los subitems. A continuación se mostraran las fases y su subdivisión.

FASE I: Diagnostico del Problema

- Contexto del Problema.
- Determinar los Objetivos a alcanzar.

FASE II: Identificación de un Modelo Matemático

- Definir una estrategia matemática de solución.
- Definir Parámetros y Variables.
- Determinar la Función Objetivo.
- Determinar las restricciones.
- Determinar los algoritmos que faciliten la solución del problema.

FASE III: Obtención de una Solución a partir del Modelo

- Adecuar un algoritmo que solucione el problema en tiempos computacionales adecuados utilizando un lenguaje de programación apropiado, de tal manera que pueda ser implementado en una herramienta computacional como Matlab.
- Obtener resultados del programa.

FASE IV: Prueba del Modelo de solución.

- Adicionar diferentes restricciones al problema de ruteamiento de vehículos.
- Examinar de manera exhaustiva que todas restricciones del modelo se estén cumpliendo.
- Realizar las corridas necesarias, hasta disminuir el error de los métodos iterativos comparados con valores exactos propuestos en la literatura especializada.

FASE V: Establecimiento de Controles sobre la Solución

- Determinar efectos de perturbaciones en los algoritmos de solución del problema.

FASE VI: Validación de la Metodología de Solución

- Ejecutar la metodología de solución y compararla con la solución obtenida en los casos de la literatura especializada.

2. PROBLEMA DE RUTEAMIENTO CON DEPENDENCIA DEL TIEMPO TDVRP

El problema específico TDVRP es una variante del problema VRP, aumentando la restricción del tiempo en el que se genera el recorrido de los vehículos. Para comprender este problema se indican los conceptos básicos del problema de ruteo desde sus inicios hasta la variante específica en el proyecto, el histórico para el planteamiento de las variantes del VRP y los métodos de solución.

2.1 ANTECEDENTES

En la literatura especializada se encuentran evidencias de artículos relacionados con el problema de ruteo desde los años 1975 hasta la actualidad. A partir del año 1989 hubo un aumento importante de publicaciones. A medida que pasa el tiempo, la investigación sobre ruteo de vehículos avanza y la cantidad de artículos publicados aumenta.

El primer problema planteado tipo VRP fue el del agente viajero o TSP (*Travelling Salesman Problem*) introducido por Flood en 1956, como lo indica L. B. Rocha Medina, et al, en [4]. Donde un agente debe visitar cierta cantidad de ciudades en un solo viaje, de tal manera que inicie y termine su recorrido en la ciudad “origen”; el agente debe determinar la ruta que debe seguir para visitar cada ciudad una sola vez y regresar de tal manera que la distancia total recorrida sea mínima.

De la formulación propuesta por Flood, nacen variaciones como el TPS generalizado en 1959 con Dantzing y Ramser; trabajo en el cual se modela el despacho de combustible a través de una flota de camiones a diferentes estaciones de servicio, desde una terminal.

En los años 1960s se encuentra la primera referencia del TSP múltiple o m-TSP con Miller, Tucker y Zemlin. Este problema es una generalización del TSP en la cual se tiene un depósito y m vehículos, es decir m agentes viajeros. El objetivo planteado es construir exactamente m rutas, una para cada vehículo, de modo que cada cliente sea visitado una vez por uno de los vehículos.

En 1969, a partir del trabajo de Tillman, se da origen al TSP probabilístico o PTSP. El objetivo de este problema es encontrar el mínimo costo de recorrido esperado a través de un conjunto de nodos con probabilidades asociadas a la presencia o no de consumidores que requieren ser servidos.

La primera consideración a observar es que el TSP generalizado, es decir el problema de los m agentes viajeros puede ser asumido como un VRP y aún más allá, como un CVRP

(*Capacited VRP*), es decir como un problema de ruteo de vehículos donde la capacidad de la flota se convierte en restrictiva para la formulación. La función objetivo del CVRP minimiza el costo total para surtir a todos los consumidores. En este tipo de problema, a los consumidores les corresponde una demanda determinística, todos los vehículos son iguales y salen de un centro de distribución.

El CVRP encuentra una cantidad exacta de rutas con mínimo costo definido como la suma de los costos de los arcos pertenecientes a los recorridos, de tal manera que cada recorrido visita el centro de distribución, cada centro de consumo es visitado por exactamente un vehículo y la suma de las demandas de los centros de consumo visitados no excede la capacidad del vehículo. Cuando el costo de ir de un centro de consumo i a otro centro de consumo j es igual al costo de ir del centro de consumo j al centro de consumo i el problema es llamado CVRP Simétrico, en caso contrario se denomina CVRP Asimétrico.

Posteriormente se habla del VRP homogéneo, refiriéndose a características comunes en las que todos los nodos manejan el mismo recurso como distancia, ventanas de tiempo, retornos y entregas fraccionadas.

El VRP heterogéneo se refiere a componentes desiguales en las que cada nodo maneja recursos distintos bien sea flota de vehículos, depósitos, viajes y componentes estocásticos en algunos casos.

En 1967 aparece el VRPTW, es decir el VRP con ventanas de tiempo, el cual presenta las variaciones tales como el VRPTD (*VRP with time deadlines*, ventanas rígidas de tiempo) en 1986, el VRPMTW (VRP con ventanas de tiempo múltiples) el cual es formulado en 1988; y VRPSTW (*VRP with soft time windows*, ventanas blandas de tiempo) en 1992.

En el artículo de C Malandraki y M. Daskin [5], se resuelve el problema de ruteamiento de vehículos considerando las condiciones temporales usando vehículos con capacidad fija y un solo depósito, donde las rutas dependen directamente de las condiciones de tiempo diarias y la solución se genera a partir de la heurística del vecino más cercano y la solución del problema del agente viajero dependiente del tiempo usando planos de corte, concluyendo que se puede usar planos de corte para un número pequeño de clientes con soluciones óptimas en este caso se llega a soluciones para 25 clientes, para el caso del vecino más cercano se generan soluciones no óptimas pero para un número más grande de clientes.

Los autores A.V Donati, et al.[6] solucionan el problema de ruteamiento de vehículos con dependencia de tiempo usando colonia de hormigas, las restricciones de tiempo se hacen con base en información tomada de casos reales de contaminación, tráfico y velocidad vehicular para obtener las penalizaciones correspondientes al tiempo de cada ruta, a esta información se documenta en un sistema denominado Cartesio. El problema se resuelve tomando 1522 puntos de coordenadas donde se encuentran los clientes geo referenciados con 2579 arcos y con una ventana de tiempo igual a media hora, además de considerar dos tipos de ruta, una que corresponde a calles con velocidad de flujo vehicular constante y las que dependen del tiempo en horarios de 8 am a 6 pm dividiendo sus periodos de tiempo de salida de los vehículos en 4 intervalos de 2,5 horas cada uno y se genera la solución para una muestra de 30 clientes de los 60 posibles extraídos aleatoriamente. Los resultados muestran un error del 7.58% en vehículos con velocidad constante, en el caso de las rutas que dependen del tiempo se indica un error del 12%. Comparado con el valor óptimo de 30 clientes.

A L. Kok, et al [7] trabajan el problema de ruteamiento con dependencia de tiempo evitando usar rutas con mayor congestión vehicular, la solución se realiza por medio del algoritmo de Dijkstra modificado y la heurística de programación dinámica restringida. Se muestra que el 99% de llegadas tarde en la visita a los clientes se puede evitar sacando las rutas que presentan mayor congestión estratégicamente, logrando mejorar el tiempo de llegada a un 70%.

Las pruebas se realizan para 15, 50 y 100 clientes usando instancias del VRP con alta reducción de llegadas tarde evitando retrasos en las visitas a los diferentes clientes y reduciendo el número de rutas y camiones.

En el artículo de H.-K. Chen, et al [8], toman el problema de ruteamiento de vehículos dependientes de tiempo en tiempo real, como un modelo de programación entero mixto para 25 clientes, dividiendo el problema a varios subproblemas de VRP con ventanas de tiempo, logran reducir el tiempo de espera además de adicionar la condición de tiempos de llegada fijos con datos de tiempos reales.

Según M. Jaime, and E. Palacios en [9] se toma el problema del ruteo de vehículos dinámico con ventanas de tiempo en tiempo real con respecto a las condiciones de oferta y demanda. Inicialmente pretende cuantificar los beneficios del uso de tiempos de viaje dependientes del tiempo en la construcción de rutas de despacho vehiculares; luego cuantificar los beneficios del uso de información en tiempo real para actualizar planes en

la operación de rutas de despacho vehiculares, considerando cambios en los niveles de servicio de la red de transporte, y nuevas demandas; y posteriormente desarrollar un algoritmo de solución para el problema.

El problema es formulado como un modelo de programación lineal entera mixta, distinguiendo dos versiones: una para el inicio del período de planificación, y otra para la actualización de las rutas, lo que ocurrirá si aparece una nueva demanda y/o si existe algún incidente que varíe los tiempos de viaje de la red. Posteriormente, se desarrolla un algoritmo de solución del problema de carácter heurístico a fin de obtener respuestas en tiempo real probando cuatro estrategias de operación que dependen del tipo de congestión considerada al momento de construir las rutas. Además, se consideraron cuatro escenarios que se distinguen por la proporción de clientes que aparecen en tiempo real, y por el nivel de impacto de los incidentes. Los resultados mostraron que, al planificar las rutas considerando ambos tipos de congestión, los costos totales disminuyen, aun visitando un número mayor de clientes.

En la tesis doctoral de Y. V. Piqueras[1] se presenta un modelo económico de distribución de mercancías que generalice los problemas de rutas sometidos a restricciones temporales de servicio y un conjunto de técnicas heurísticas y metaheurísticas capaces de resolverlo eficientemente, que en este caso se nota mejores resultados con algoritmos genéticos. Para ello se define una función objetivo que mide la rentabilidad económica de las operaciones, y flexibiliza los horarios de entrega con penalizaciones que reflejen la insatisfacción de los clientes. Asimismo se contempla la posibilidad de contar con flotas heterogéneas de vehículos con costes fijos y variables diferenciados, así como capacidad de carga, velocidad y jornadas laborales distintas, y con la posibilidad del uso múltiple. Se incorpora la asimetría en la duración de los viajes, con tiempos de aproximación y de alejamiento que modulen el nivel de congestión por tráfico y otras dificultades de acceso. También es posible el ajuste de diferentes costes horarios en función de las horas extraordinarias y penalizaciones por ruptura en la llegada al depósito.

M. R. Khouadjia, et al. [10] hablan de la solución del problema de ruteo de vehículos con ventanas de tiempo por medio de técnicas heurísticas de enjambres, las cuales comparan el rendimiento de la solución con la heurística de la búsqueda en vecindarios variables y se evalúa de acuerdo a la adaptación dinámica que tengan.

En el artículo de N. Mladenović, et al. [11], Los autores resuelven el problema del agente viajero como un vehículo que sale del depósito y lleva un producto a los clientes y además puede hacer entrega y recogida del mismo, para ello usan diferentes heurísticas de

intercambio como el *k-opt* y búsqueda en vecindario variable; las respuestas que se plantean están en función de la solución del problema usando instancias de 200, 500 y hasta 1000 clientes.

Los autores S. Kritzinger, et al. [12] resuelven el problema de ruteo de vehículos dependientes del tiempo por medio de la recolección de información real y mediante el uso del algoritmo de Dijkstra, además se adiciona la restricción de ventanas de tiempo, para ello se resuelve usando la búsqueda en entornos variables aplicando dichas soluciones en las calles de Viena.

Los autores K. Fleszar, et al. [13], resuelven el problema de enrutamiento abierto de vehículos (OVRP), que tiene como objetivo, minimizar el número de vehículos y por consiguiente la distancia total de recorrido. Visitando un número de clientes y regresando al depósito, de tal forma que se cumpla la demanda de los clientes con un solo vehículo, sin que exceda su capacidad. El problema se resuelve de manera eficiente utilizando el VNS.

Según el autor J. F Delgado O[14], se presenta un modelo de programación entera para el problema RoSLoP (*Routing- Scheduling-Loading Problem*) en una compañía caso de estudio. La meta es obtener una solución para el proceso de entrega de productos embotellados, minimizando el número de vehículos y la distancia de recorrido de los vehículos para satisfacer los pedidos de los clientes. Para ello se ha formulado a partir de dos problemas clásicos de optimización combinatoria: el problema de ruteamiento de vehículos y el problema de empaqueo de objetos, típicamente, los investigadores resuelven problemas de ruteamiento de vehículos basados en problemáticas reales de transportación de bienes modelando un número mínimo de variantes del problema VRP. Simplificando la complejidad del problema, ante la limitante de métodos de solución exactos que permitan obtener una solución óptima para problemas de optimización clasificados por su complejidad.

Para establecer el grado de exactitud de los métodos aproximados, por lo cual se desarrolló un método exacto. El cual, utiliza una función de transformación lineal para la representación de las unidades de asignación y la dimensiones de los vehículos, lo cual, permite la reducción del espacio de soluciones y la solución óptima para problemas de hasta 12 variantes de VRP como un problema *rich VRP*. Estos resultados experimentales del método exacto demuestran que los métodos aproximados obtienen un 98% de aproximación en la calidad de la soluciones. El método exacto obtiene una solución

óptima para el problema *rich VRP* asociado en un tiempo promedio de 4 segundos, obteniendo un ahorro del 94% respecto al criterio de paro de los métodos aproximados. Dada la definición del problema, y dada la formulación como un problema de programación lineal entera mixto, se concluye que se ofrece una solución factible de buena calidad a la compañía caso de estudio.

Los autores A. Imran, et al. [15], muestran cómo resolver el problema de ruteo de vehículos con flota heterogénea usando la heurística de intercambios y la búsqueda en entornos variables, los resultados indican una reducción en un 13 % del tiempo de computo comparado con el algoritmo de Dijkstra usando instancias de hasta 100 clientes.

Según E. Stegers [16], en su tesis propone desarrollar la solución del problema de ruteo de vehículos dependientes del tiempo tratando de obtener un acercamiento con el modelo presentado en la realidad, motivado por los cambios dinámicos en el tráfico vehicular, para cumplir con el objetivo propone inicialmente minimizar el número de vehículos usados en recorrer los nodos o clientes, para luego pasar a minimizar los costos del problema.

2.2 MARCO REFERENCIAL

El problema de ruteo de vehículos es un elemento fundamental en la cadena de suministros puesto que la garantía de entrega de productos a los clientes en tiempos adecuados y con las condiciones deseables del producto hacen que las empresas tengan un buen posicionamiento en el mercado y por ende presenten incrementos en sus ganancias, es por esto que el ruteo de vehículos permite dar un aporte importante en la reducción de costos al momento de realizar las entregas de los productos a los diferentes clientes.

El interés de este trabajo es resolver el problema desde la óptica del TDVRP porque es más cercano a la situación real donde se considera el ruteo teniendo en cuenta que la empresa de distribución de productos lácteos en la ciudad de Pasto actualmente tiene cerca de 2500 clientes, a los cuales les lleva sus productos usando 24 rutas con vehículos de capacidad fija y homogéneos, los cuales salen en un horario preferido por los clientes por razones comerciales entre las 6 y las 7 de la mañana. Los vehículos se encuentran en este horario con el problema de la hora pico en la cual la densidad vehicular en la ciudad es más alta, lo cual conlleva a tomar rutas alternas para cumplir con las entregas a los diferentes clientes.

Para contextualizar el estado del arte del problema se realizó una revisión bibliográfica del problema donde se evidenció que no aparecen metodologías similares a la propuesta en

este trabajo, pero que sirven como referencia y apoyo a la implementación algorítmica de la solución del problema.

En la Tabla 2-1 se muestra un resumen de los artículos revisados, en los que se observan aspectos tales como la metodología de solución, la forma como se realiza una comparación de resultados, el número de instancias de prueba y principalmente metodologías de implementación y codificación algorítmica, además de restricciones en los diferentes problemas que resuelven y sus límites.

A continuación se presenta un compendio de artículos especializados que enmarcan las temáticas que se emplearán en el desarrollo de esta investigación, cabe resaltar que la temática específica de ruteo de vehículos con dependencia de tiempo es un problema poco explorado y las técnicas descritas en la literatura especializada son diferentes a la que se pretende implementar, sin embargo los siguientes artículos relacionados permiten comprender el problema y algunas de las herramientas para su solución.

La columna "*No de clientes*" indica la cantidad de clientes o rutas para las que se resuelve el problema en cada caso.

La columna de "*Considera demanda*" muestra para los diferentes problemas el hecho de considerar la demanda de los clientes o no.

La columna de "*Pruebas*", indica las instancias que los autores usan para calibrar y comparar resultados para diferentes condiciones o si el problema se resuelve con valores obtenidos de casos de aplicación real.

Tabla 2-1. Revisión del estado del arte.

Referencia de artículo	Problema	Método de solución	Codificación	No de clientes	Considera demanda	Pruebas
The real-time time-dependent vehicle routing problem [8]	TD VRPTW	Programación entera mixta	Branch-Bound Inserción	36	Si	56 Instancias de Solomon
Time dependent vehicle routing problem with a multi ant colony system. [6]	TD VRP	Búsqueda local, hormigas, algoritmo de la ruta más corta.	Probabilidad de evaporación de feromona.	80	Si	Aproximación de caso real.

TESIS DOCTORAL Optimización heurística económica aplicada a las redes de transporte del tipo VRPTW. [1]	VRPTW	Algoritmo de las economías, basado en el uso de varias Meta heurísticas.	Dependiente de la heurística y Meta heurística. Se usan algoritmos básicos y codificación binaria para genéticos y VNS	19 rutas	Si	Instancias
Modelado matemático para el análisis y mejoramiento de los métodos aproximados para la solución de los problemas de enrutamiento, asignación de horarios y distribución de cargas: una aplicación a la entrega de productos embotellados. [14]	VRP y BPP	Programación entera	Modelo matemático. Usando SW LINDO API 4.0	6 vehículos	Si	Base de datos empresa
Vehicle routing under time-dependent travel times: The impact of congestion avoidance. [7].	TD VRP	Algoritmo de Dijkstra Programación dinámica	Modelo matemático Solución de varios TSP Medir velocidad de flujo y hora, considerando velocidad limite.	15, 50 y 100 clientes	No	Instancias y casos reales en carretera de España
Time Dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms. [5]	TD VRP	Modelado matemático. Probabilidad de tráfico según el horario	TSP generalizado a VRP usando programación lineal entera mixta Planos de corte	25 clientes	No	Instancias de Christofides
Application Heuristic, Exact and Hybrid Approaches for Vehicle Routing Problems. [17]	VRP, CVRP, VRPPD,VRP flota heterogénea	Modelo matemático.	Método de ramificación y planos de corte y búsqueda local iterada	200 clientes con errores pequeños	Si	Instancias VRP
The Time-Dependent Pollution-Routing Problem. [18]	TD VRP adicionando restricción de contaminación ambienta (TDPRP)	Modelo matemático. Uso de SW CPLEX 12.1	Programación lineal entera FO: Minimizar el uso de combustible	20 nodos	no	Medida de distancias reales en ciudades al azar del Reino Unido

The Time-Dependent Vehicle Routing Problem with Soft Time Windows and Stochastic Travel Times. [19]	TD VRP	Búsqueda Tabú, VNS.	1. Construcción de solución inicial. 2. Mejorar usando TS. 3. Mejorar con VNS	100 clientes	no	Instancias de Salomón
An Ant Colony algorithm hybridized with insertion heuristics for the Time Dependent Vehicle Routing Problem with Time Windows [20]	TD VRP TW	Hormigas más algoritmo de inserción (Movimiento de dos o más aristas 2 OPT, Cross)	Actualización de feromona. Intensificación de búsqueda con algoritmos clásicos de intercambio	90 clientes	no	Instancias de Salomón.
A dynamic vehicle routing problem with time-dependent travel times [21]	TD VRP TW dinámico	Algoritmos Genéticos. Corridas dinámicas según la hora de salida del vehículo	Codificación entera FO: suma de costo por distancia mas hora de salida y entrega y recogida de producto	20 rutas con 200 clientes y horario de 8 am a 6 pm	no	Instancias desconocidas
Optimizing goods assignment and the vehicle routing problem with time-dependent travel speeds [22]	TD VRP	Modelo matemático	Series de varios problemas de programación entera mixta	210 clientes	no	Casos reales de ciudades de Japón
Computing nine new best-so-far solutions for Capacitated VRP with a cellular Genetic Algorithm. [23]	C VRP	Algoritmo genético con intensificación por medio de búsqueda local	Codificación binaria. 2 opt	150 clientes	sí	Instancias de Christofides
Integrated GA-VRP solver for multi-depot system [24]	VRP con multiple depósito	Algoritmo genético	Codificación entera con dos puntos de cruce en cada padre.	99 clientes	no	Instancias para TSP de Kureichick
A hybrid genetic – Particle Swarm Optimization Algorithm for the vehicle routing problem [25]	VRP	Algoritmo genético y enjambres	Conservación de padres para sacar de óptimos locales al algoritmo genético	No definido	no	Instancias VRP

A vehicle routing problem solved by using a hybrid genetic algorithm. [26]	VRP	Algoritmo genético	Codificación entera. Generación de población inicial con heurísticas de inserción	No definido	no	El interés de los investigadores está en adecuar una función de probabilidad para la mutación, lo cual permite salir de óptimos locales
A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. [27]	VRP con entrega y recogida	Algoritmo genético	Codificación entera	30 a 44 clientes	no	Instancias VRP
Este proyecto	TD VRP	Algoritmo genético. Heurísticas VNS	Codificación entera	160 clientes	si	Instancias VRP, Caso de aplicación real. Generación de matriz de costos real. Generación de dependencia del tiempo real

2.3 MARCO CONCEPTUAL

2.3.1 Problemas característicos en la distribución y transporte.

Los problemas de distribución consisten básicamente en asignar una ruta a cada vehículo de una flota para repartir o recoger mercancías. Los clientes se localizan en determinados puntos o arcos y a su vez pueden tener horarios de servicio determinados. como lo indican H. K. Chen, et al en [8].

Estos problemas pueden tener diferentes formas de caracterización así:

- Por el tamaño de la flota: Se puede considerar un solo vehículo, el número de vehículos limitado o el número de vehículos ilimitado.
- Por el Tipo de flota: Esta puede ser de un solo tipo de vehículo o homogénea, o con múltiples tipos de vehículos o heterogénea; también se puede usar vehículos con compartimentos para las mercancías o sin compartimentar.

- Por el depósito o almacén: Se puede considerar depósito único, múltiples depósitos y depósitos de reaprovisionamiento intermedios.
- Por Naturaleza de la demanda: que puede ser determinada o estocástica, también puede depender del tiempo o dinámica, puede ser satisfecha parcialmente o puede ser fija para todos los clientes como también puede ser variable.
- Localización de la demanda: La demanda puede estar tanto en los nodos como en los arcos o ser mixta.
- Restricciones horarias de servicio: Se puede considerar sin restricciones, también en el inicio o en el final, como en ambos casos, pueden ser estrictas o flexibles con penalización, o pueden tener ventanas temporales únicas o múltiples.
- Ciclos de servicio: Se puede realizar una sola vez en un ciclo o varios recorridos.
- De acuerdo al grafo que representa: Puede ser orientado con una sola dirección, no orientado sin dirección, puede ser mixto y Euclídeo, es decir con todos los vértices conectados a un nodo.
- Restricciones en la capacidad de los vehículos: La capacidad puede ser limitada y única para todos o limitada y diferente para algunos vehículos.
- Velocidad media de los vehículos: Podemos considerar una velocidad constante, dependiente del horario, del trayecto o de los dos.
- Duración máxima de una ruta: se puede considerar limitada e igual para todas, diferente, estricta o flexible.
- Número de rutas por vehículo: Pueden ser una para cada vehículo, también se pueden considerar varias rutas para cada vehículo.
- Operaciones: En este caso se refiere al reparto de las mercancías, la recogida o los dos casos.
- Relaciones de precedencia: puede o no existir precedencias tanto para clientes como para uso de vehículos.
- Mercancías. Puede tratarse de productos iguales o diferentes.
- Costo: El costo puede ser fijo o variable o mixto, también se puede considerar penalizaciones por demandas insatisfechas.
- Función objetivo: Los modelos pueden basarse en minimizar costos fijos o variables en el recorrido, es decir la distancia o también se puede enfocar a la minimización de rutas o vehículos a usar, minimizar tiempo empleado, minimizar esperas, maximizar beneficio, maximizar función de utilidad del cliente, entre otros.

Del listado anterior se tienen distintos problemas con todo un conjunto de restricciones que permiten obtener un modelado que corresponde a un problema difícil de resolver por el gran número de combinaciones posibles, sin embargo no se están considerando otras restricciones del mundo real como pueden ser el clima, el tráfico vehicular, la congestión

vehicular, normas de tránsito, entre otras que restringen aún más los modelos y por ende dificultan la búsqueda de soluciones óptimas, sin embargo con los modelos básicos se trata de generar soluciones confiables y cada vez aproximar mejor a las condiciones de la realidad del problema.

De acuerdo a esto se relaciona a los problemas desde el más básico como es el agente viajero o TSP "*Traveling Salesman Problem*" en el que se supone un agente que recorre varias ciudades y que debe regresar al punto de partida con el objetivo de minimizar la ruta. De este problema se puede observar el gran número de combinaciones posibles con el orden de $n!/2n$ para los posibles recorridos, es decir que sería difícil tratar de resolver el problema con un algoritmo polinomial para una solución exacta.

Otro problema semejante es el m-TSP "*Multiple Traveling Salesmen Problem*", donde existen m agentes para cubrir las diferentes ciudades y que cada ciudad sea visitada por un único agente.

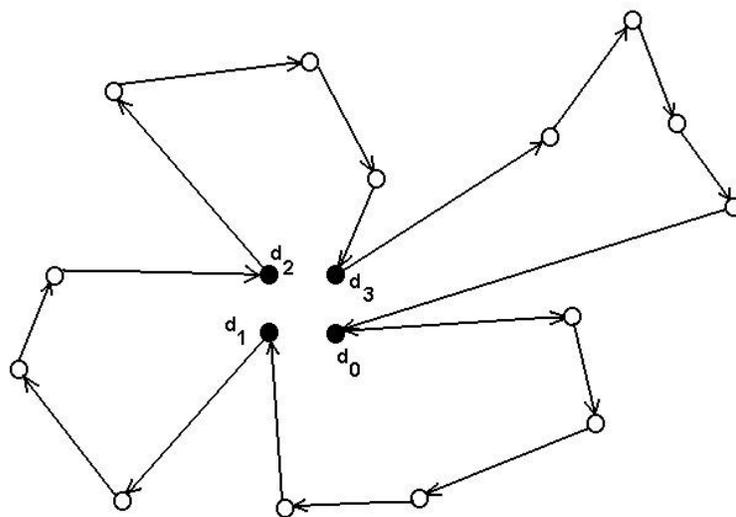


Figura 2-1. Esquema problema m-TSP

Un problema semejante es el del cartero chino "*Chinese Postman Problem*" (CPP), formulado por Euler en el siglo XVIII, constituye uno de los ejemplos donde la demanda está repartida a lo largo de un arco. En efecto, dado un grafo conexo, se trata de cubrir todas las aristas minimizando la distancia recorrida (permitiéndose recorrer los arcos varias veces). El problema clásico donde los arcos no están orientados, puede resolverse mediante algoritmos polinomiales.

Posteriormente se tiene un problema más específico al incluir rutas en los recorridos desde un almacén a las diferentes ciudades, denominándose así al VRP o problema de ruteamiento de vehículos "*Vehicle Routing Problem*", es una generalización del *m-TSP*, donde existe una demanda asociada a cada ciudad y una capacidad determinada de transporte para cada uno de los vehículos.

Cuando se tiene la capacidad de carga de los vehículos se puede definir un problema llamado CVRP "*Capacitated Vehicle Routing Problem*", debido a que el total de la demanda de las ciudades visitadas en una ruta no debe superar la capacidad del vehículo. El objetivo puede ser reducir al mínimo posible la suma de la distancia recorrida por todas las rutas, el número de vehículos, o una combinación de ambos criterios. Cabe notar que la dirección de la ruta no es una restricción al problema.

Este problema presenta algunas variantes que tratan en lo posible de asemejar a la realidad aumentando restricciones a su modelamiento. Algunos de estos se listan a continuación:

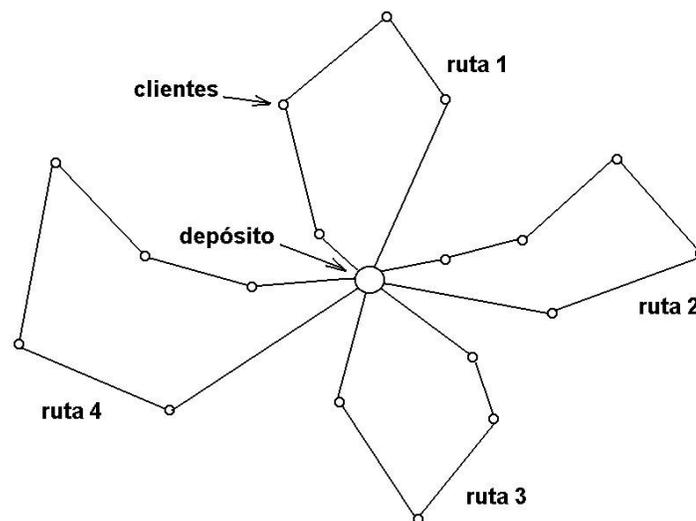


Figura 2-2. Esquema del problema de ruteamiento de vehículos VRP.

1. Problema de ruteamiento de vehículos asimétrico AVRP: En este problema el costo del trayecto depende del sentido.
2. Problema de ruteamiento de vehículos con capacidad fija CVRP: La capacidad de transporte del vehículo debe ser fija y no se puede exceder en la demanda de las ciudades visitadas.

3. Problema de ruteamiento de vehículos con restricción de longitud VRPLC: La longitud de cada ruta no debe superar una magnitud determinada. Alternativamente se puede limitar el tiempo.
4. Problema de ruteamiento de vehículos periódico PVRP: Se debe satisfacer la demanda del cliente en un periodo determinado de tiempo.
5. Problema de rutas fijas FRP: Las rutas no cambian durante un periodo de tiempo aunque cambie la demanda.
6. Problema de ruteamiento de vehículos con flota heterogénea VRPHE: Los costos pueden ser fijos o variables que dependen del tipo de vehículo, aunque no existe límite de vehículos.
7. Problema de ruteamiento de vehículos con recogida de regreso VRPB: Se puede hacer entregas y luego de estas permitir a los vehículos recoger los productos.
8. Problema de ruteamiento de vehículos con entrega y recogida de regreso VRPDB: Existen puntos de entrega y otros de recogida hacia el almacén, pudiendo coincidir en ambos. Se permite la entrega y recogida mientras no se viole la capacidad del vehículo.
9. Problema de entrega y recogida PDP: En este problema un mismo vehículo debe llevar la mercancía de un sitio a otro en toda la red.
10. Problema de ruteamiento de vehículos con múltiple compartimento: Se debe transportar varios tipos de mercancías separadas durante todo el viaje.
11. Problema de ruteamiento previo de vehículos VRPPC: Antes de visitar a un cliente se debe visitar a un conjunto previo.
12. Problema de ruteamiento de vehículos con múltiple depósito y capacidad fija MCVRP: En este problema los vehículos tienen la misma capacidad y existen varios depósitos.
13. Problema de ruteamiento de vehículos con reabastecimiento VRPSF: Existen depósitos intermedios donde los vehículos pueden reabastecerse.
14. Problema de ruteamiento de vehículos abierto OVRP: No se requiere que los vehículos regresen al depósito.
15. Problema de ruteamiento de vehículos dinámico DVRP: Algunos parámetros del modelo dependen del tiempo.
16. Problema de ruteamiento de vehículos con tiempo de viaje variable VRPVRT: La duración de los viajes dependen del horario en el que se realicen.
17. Problema de ruteamiento de vehículos con demanda estocástica VRPST: Los viajes tienen una duración aleatoria que puede ser extendida a los clientes, es decir que pueden o no estar en ciertos horarios.
18. Problema de ruteamiento de vehículos de uso múltiple VRPM: En este caso los vehículos pueden ir a distintas rutas.
19. Problema de ruteamiento de vehículos con entrega dividida VRPSDV: La demanda de un cliente puede ser cubierta por varios vehículos.
20. Problema de ruteamiento de vehículos con ventanas de tiempo VRPTW: Cada cliente tiene un horario establecido para la entrega de la mercancía.

21. Problema de ruteamiento de vehículos dependientes de tiempo VRPTD: El horario de entrega sólo tiene como limitación su final, y los recorridos en las rutas dependen del estado de las vías, el tráfico y el tiempo en el que se hace el recorrido.

En general cualquier tipo de problema mencionado anteriormente es de combinación exponencial y computacionalmente muy complejo para resolverse de forma exacta, por ello es importante aplicar algoritmos de aproximación que proporcionen soluciones viables y eficientes.

2.3.2 Modelo matemático del problema de ruteo de vehículos con dependencia del tiempo

El modelo se formula generalmente estableciendo las variables de decisión o control, cuyos valores finales solucionarán el problema, se determina una función objetivo, expresada en términos de los datos y de las variables de decisión y fijando límites o restricciones de los valores de las variables debido a las condiciones externas del problema. La solución a este tipo de modelos se realiza por medio de herramientas matemáticas que comprenden la programación matemática lineal, entera, no lineal, dinámica, el análisis determinista, la teoría de la probabilidad o estocástica, la teoría de inventarios, los fenómenos de espera y la simulación. Los cuales presentan técnicas para la solución de los modelos matemáticos usando métodos exactos que producen soluciones óptimas en la medida que el problema lo permita para así cumplir con cada una de las restricciones, sin embargo, cuando el problema genera gran cantidad de combinaciones, los procesadores no tendrían el tiempo suficiente para resolver el problema completamente por lo que se plantean técnicas heurísticas que proporcionan opciones viables cuasi-óptimas y útiles.

El modelado inicia con la formulación matemática para un problema básico como es el TSP, al cual posteriormente se adicionan restricciones para formar el modelo del VRP y sucesivamente de acuerdo con las restricciones que se tengan se forman los diferentes modelos de ruteamiento tales como capacidad homogénea, tipo de flota, ventanas temporales etc.

Modelo Matemático Para El TSP.

En el problema del cartero viajante – Traveling Salesman Problem o TSP, existe un conjunto de n ciudades (nodos), $V = \{1, 2, 3, \dots, n\}$, y un conjunto de caminos (arcos) uniendo cada una de las ciudades, así el camino $(i,j) \in A$, c_{ij} es la “distancia” (función objetivo) para ir de la ciudad i a la ciudad j (c_{ij} no necesariamente es igual a c_{ji}). Un cartero debe realizar un tour (recorrido) comenzando en una cierta ciudad de origen y luego

visitar todas las otras ciudades una única vez y retornar a la ciudad de origen. El problema consiste en hallar el tour (recorrido) de distancia mínima evitando subtours.

El problema del TSP tiene el siguiente modelo matemático, mostrado por R. A. Incapíe, et al, en [28]:

$$\text{Minimizar } z(\mathbf{x}) = \sum_{i,j \in A} c_{ij} x_{ij} \quad (2-1)$$

Sujeto a:

$$\sum_{i: i,j \in A} x_{ij} = 1 \quad \text{para todo } j \quad (2-2)$$

$$\sum_{j: i,j \in A} x_{ij} = 1 \quad \text{para todo } i \quad (2-3)$$

$$\sum_{i,j \in A} x_{ij} \geq 1 \quad (2-4)$$

$$\sum_{i,j \in A} x_{ij} \leq |S| - 1, \text{ para todo } S \in V, |S| > 1 \quad (2-5)$$

$$x_{ij} \in \{0, 1\} \quad (2-6)$$

El modelo indica la función objetivo (2-1) dado por el costo de ir de un punto x a otro. La *restricción* (2-2) permite visitar una única vez a cada cliente. La *restricción* 2-3 permite que todos los clientes sean visitados. La *restricción* (2-4) indicando que las variables de decisión sean de tipo binario, es decir que se puede o no visitar a un cliente. La *restricción* (2-5), indica que no existan sub-tours para cada recorrido S , en la visita de un vehículo sobre los diferentes clientes, sin permitir ciclos infinitos entre ellos. La *restricción* (2-6), muestra como la variable debe ser entera.

Modelo Matemático Para El CVRP.

En una siguiente instancia se puede obtener el modelo matemático para el problema general del ruteo de vehículos de la siguiente manera:

Como objetivo principal está la disminución del costo de recorrido de los vehículos en todos los trayectos.

Posteriormente se tiene un número máximo de vehículos con capacidad homogénea fija, los cuales salen desde un punto denominado almacén, el cual será el origen de las diferentes rutas que recorren todos los vehículos, cubriendo todos los puntos o clientes, los cuales presentan una demanda conocida que debe ser satisfecha por el único vehículo que lo visite. Los índices y variables del problema son:

i : nodo de partida del vehículo.

j : nodo de llegada del vehículo.

k : Vehículo a utilizar (1,2,3,... k)

d : Demanda del cliente.

x_{ij}^k : Si es igual a 1, el vehículo k es asignado para recorrer el arco desde el nodo i al nodo j . Es igual a 0 en caso contrario.

y_{ij} : 1 si se realiza el recorrido desde el nodo i , hasta el nodo j . 0 en caso contrario.

Se pueden definir otros elementos como parámetros así:

c_{ij} = Costo de transporte del nodo i al nodo j

d_j = Demanda en el nodo j

u_k = Capacidad del recurso k

n = Cantidad de clientes

El Modelo matemático de ruteo según los aportes de Ahuja et al. [29] y A. Olivera [30], se lista con las ecuaciones siguientes:

$$\text{Minimizar } \sum_{(i,j) \in A} c_{ij} * y_{ij} \quad (2-7)$$

Sujeto A:

$$\sum_{1 \leq k \leq K} x_{ij}^k = y_{ij}; \forall i, j \quad (2-8)$$

$$\sum_{1 \leq j \leq n} y_{ij} = 1 ; \forall i \quad (2-9)$$

$$\sum_{1 \leq i \leq n} y_{ij} = 1 ; \forall j \quad (2-10)$$

$$\sum_{1 \leq j \leq n} y_{0j} = k \quad (2-11)$$

$$\sum_{1 \leq j \leq n} y_{i0} = k \quad (2-12)$$

$$\sum_{1 \leq i \leq n} \sum_{1 \leq j \leq n} d_i * x_{ij}^k \leq u ; \forall k \quad (2-13)$$

$$\sum_{i \in Q} \sum_{j \in Q} y_{ij} \leq |Q| - 1 ; \forall \text{ subconjunto de } Q \text{ de } \{1, 2, \dots, n\} \quad (2-14)$$

$$k \leq K \quad (2-15)$$

$$y_{ij} \in \{0, 1\} ; \forall (i, j) \in A \quad (2-16)$$

$$x_{ij}^k \in \{0, 1\} ; \forall (i, j) \in A, \forall k \quad (2-17)$$

El conjunto A se define como: $A = \{(i, j) : y_{ij} = 1\}$. Es decir el conjunto de aristas del grafo con la posibilidad de hacer un solo recorrido desde el nodo i hacia el nodo j . Todas las rutas empiezan en 0 y terminan en 0 . Cada arco $(i, j) \in A, i \neq j$, de la red tiene asociado un costo c_{ij} . Cada vehículo se caracteriza por tener una capacidad de carga q . Cada cliente presenta una demanda $d_i, i \in C$. Para cada uno de los clientes, También se asumirá que todos los datos son números enteros conocidos y no negativos. De este modo, el modelo es determinístico.

Las restricciones presentadas en el modelo tienen como finalidad diseñar un conjunto de rutas de costo mínimo, una para cada vehículo, de forma que:

- Se atienda exactamente una sola vez a cada cliente.
- Cada ruta empiece y termine en el depósito.
- Se respeten las restricciones de capacidad de los vehículos.

En el modelo encontramos inicialmente la función objetivo en (2-7), la cual indica que se requiere minimizar la suma de los costos de ir desde el depósito hacia todos los nodos, donde el costo básicamente es la distancia entre dichos nodos.

Se debe asignar a cada cliente un vehículo y una secuencia de clientes para cada vehículo que logre el costo mínimo.

La *restricción (2-8)* permite obligar la asignación de un vehículo a la ruta (i,j) , si esta es recorrida, y no asignarlo si la ruta no se va a recorrer, esta restricción contiene la variable de decisión x que indica que si $x=1$, se usa el vehículo k en el arco i,j o no.

Las *restricciones (2-9) y (2-10)*, indican la activación del arco i,j por medio de la variable y , lo que determina un recorrido entre los nodos i,j , además se asegura que todo cliente es un nodo intermedio de alguna ruta. Es decir que se asegura que cada cliente es visitado una sola vez por un vehículo.

Las *restricciones (2-11) y (2-12)*, indican que k es la cantidad de vehículos utilizados en la solución y que todos los que parten del depósito deben regresar al mismo.

La *restricción (2-13)* observa que cada vehículo no sobrepase su capacidad.

La *restricción (2-14)* vigila que la solución no contenga ciclos usando los nodos $1,2,\dots,n$. De otra manera los arcos de A contendrían algún ciclo pasando a través de un conjunto de nodos Q y la solución violaría la restricción, porque el lado izquierdo de la restricción sería al menos $|Q|$.

La *restricción (2-15)* limita el número máximo de vehículos a utilizar hasta una cantidad máxima.

Finalmente las *restricciones (2-16) y (2-17)* indican que las variables x , y son binarias.

Modelo Matemático Para El TDVRP.

Para el caso del problema específico de ruteo de vehículos con dependencia de tiempo o Time Dependent VRP, (TDVRP). Se presenta el mismo modelamiento del CVRP, con adición de la restricción donde el tiempo de transporte de un punto a otro pueden variar dependiendo de la hora en que se genere el viaje por la arista o camino que une dos clientes, esto se debe a la intensidad del tráfico que incrementa o disminuye a determinadas horas del día.

Cuando en algún caso la demanda particular de un cliente supere la capacidad del vehículo, el esquema teórico mantiene su validez simplemente creando nuevos clientes ficticios con la misma localización y horarios de servicio, pero con su demanda dividida.

Según Che, -et al. [8] en su artículo, referencian un modelamiento matemático de programación entera para el problema de ruteo de vehículos dependientes del tiempo más una restricción de trabajo en tiempo real y además considerando ventanas temporales RT-TDVRPTW. En este artículo los autores presentan el modelamiento con el objetivo de encontrar la ruta más corta del problema y con la particularidad de poder incluir nuevos clientes a una ruta en la medida en que estos lo requieran a cualquier hora del día.

De los planteamientos dados por los autores se toma la *función objetivo (2-18)* así:

$$\begin{aligned}
 \min_{\{X_{i,j}(m), w_1, w_2\} \in \Omega} z(t) \\
 = \alpha \sum_{i \in N_{<u}(t)} \sum_{j \in N_{u0}(t)} \sum_k \sum_{m > m_1 u(t)} C_{ij}(m) X_{ijk}(m) + \beta \sum_{i \in N_{u,t}} w_1 \\
 + \gamma \sum_{i \in N_{u,t}} w_2
 \end{aligned} \tag{2-18}$$

Los autores muestran al problema considerándose la suma de todos los costos al ir de un punto a otro más un factor β que afecta al costo del tiempo por causa del horario de salida y la congestión. También los autores presentan un factor γ que indica el sobrecosto de tiempo real, es decir un parámetro que debe ser testeado de acuerdo a las condiciones reales de tiempo en el que se encuentra cada vehículo transitando, afectando al costo global de forma dinámica, teniendo que resolver el problema cada cierto instante de tiempo según la capacidad del algoritmo en una región factible Ω .

Para el caso del problema a resolver en la empresa COLACTEOS, se considera una función objetivo en la que se tenga el efecto de una constante de sobrecosto dada por el horario en el que se haga el recorrido de cada ruta, es decir que se considera el factor β de la *restricción 2.18*. Pasando así a complementar la *función objetivo (2.7)*:

$$\text{Minimizar} \sum_{(i,j) \in A} c_{ij} * y_{ij} + \beta * \sum_{(i,j) \in A} c_{ij} * y_{ij} \tag{2-19}$$

Adicionalmente, según los autores Chen, et al. [8], en el modelo que plantean, consideran la restricción de capacidad de los vehículos así:

$$\sum_{i \in N_u(t)} q_i \sum_{j \in N_{u0}(t)} \sum_{m \geq m_{ij}(t)} x_{ijk}(m) \leq Q_k - \bar{Q}_k(t) \quad \forall k \in K \quad (2-20)$$

En la *restricción (2-20)*, los autores suman la demanda q_i de cada ciudad i y cada ruta j que ha sido atendida por el vehículo k y verifican que no exceda la capacidad máxima Q del vehículo k , además se resta de la capacidad total el valor de carga que se recoja en de los diferentes clientes $\bar{Q}_k(t)$.

De esta forma para el caso particular de la empresa COLACTEOS, se agrega la restricción al modelo CVRP, sin considerar recogidas de productos a los clientes, así:

$$\sum_{i \in A} q_i \sum_{j \in A} y_{ij} \leq Q_k \quad (2-21)$$

Finalmente para el caso particular de la empresa COLACTEOS, se toma como referencia el modelo del CVRP planteado anteriormente y con el complemento en la función objetivo mostrada en la *restricción (2-19)*. Además de la restricción de capacidad de carga en los vehículos dada en la *restricción (2-21)*.

2.3.3 Métodos de solución a problemas combinatoriales

Con el fin de comprender mejor las técnicas de solución existentes tanto de manera exacta como por aproximación usando heurísticas y metaheurísticas se describe algunos elementos importantes de la historia de este tipo de problemas y su complejidad computacional, para seleccionar finalmente algunas técnicas que permitan dar una respuesta eficiente y eficaz en la solución al VRP y su variante TDVRP.

Según la tesis de Y. V Piqueras [1], se indica como Jordán, Minkowsky, Farkas, Erlang y otros en los inicios del siglo XIX usan herramientas de la Investigación Operativa con trabajos sobre modelos lineales para mejorar la eficacia, efectividad y resultados de las operaciones industriales, administrativas, militares, sanitarias, económicas, entre otras. Los primeros modelos de programación lineal matemática se muestran en economía debidos a Quesnay (siglo XVIII) y Walras (siglo XIX), los que fueron perfeccionados por autores como Von Neumann, Kantorovich y Dantzig; fundamentando así la investigación operativa en la construcción de modelos de los sistemas reales y su análisis posterior por

medio del ajuste de datos experimentales, o de sus resultados para efectuar cambios en las operaciones, desarrollar y predecir comportamientos posteriores.

En la década de los setenta surge el interés por notar que tan complejos pueden ser los problemas combinatoriales y sus algoritmos de solución definiendo así el término de complejidad computacional. Notándose así el rendimiento algorítmico en la medida del tiempo medio de ejecución empleado por un procedimiento para completar su operación con un conjunto de datos estableciendo el esfuerzo de cálculo para un procesador de acuerdo a la dimensión del problema.

M. Ballou, en [31] muestran como un algoritmo presenta una complejidad polinómica P , si necesita un tiempo $O(nk)$, donde n muestra la dimensión de entrada y k es una constante independiente de n . Si la función que denota la complejidad no está acotada por un polinomio, el algoritmo presenta una complejidad en tiempo exponencial, los problemas NP son aquellos que presentan soluciones en un tiempo polinomial indeterminado para un procesador que sólo los problemas P son solucionables eficientemente mediante algoritmos, no conociéndose un procedimiento polinomial de resolución para los NP.

Las soluciones planteadas a estos problemas inician con George B. Dantzing quien en 1947 concibió el método simplex para resolver este problema con modelos de programación lineal. En 1939 el matemático y economista soviético L.V. Kantorovich planteó y solucionó un problema de programación lineal relacionado con la organización y la planificación, aunque su trabajo permaneció sin conocerse hasta 1959. En el verano de 1948, el también economista y matemático T.C. Koopmans mientras paseaba con Dantzing cerca de la playa de Santa Mónica en California. H.D. Bazarra en [32] Indica el termino de programación lineal y recopilando los trabajos de Dantzing en 1951 en el libro Activity Analysis of Production and Allocation editado por Koopmans.

La idea del método simplex se basa en recorrer el poliedro formado por las restricciones del programa lineal, vértice a vértice, a lo largo de las aristas, hasta alcanzar el vértice óptimo.

Posteriormente surge la programación entera que trata de optimizar aquellos problemas que en algunas o todas las variables de decisión se restringen a un conjunto de valores discretos. Gomory en [33] publica el primer algoritmo de programación lineal entera y en la década de los cuarenta se resolvió el problema del transporte. Para luego tratar problemas clásicos como el de la mochila, el del viajante, la programación proyectos, la localización de recursos, etc.

Algoritmos de solución exactos

Los métodos exactos para problemas de ruteo son eficientes en problemas con pocos clientes como lo indica N. Azi et al.[34].en su artículo en el que toma una variante del VRP con ventanas de tiempo, dado el tamaño del problema y algunas características del problema como la restricción de tiempo que tarda la entrega de productos en la ruta, para el cual afirma en sus conclusiones que los métodos clásicos de programación lineal entera pueden solucionar de forma rutinaria casos con 25 clientes y en algunos casos con un máximo de 50 clientes debido a restricciones de tiempo computacional.

Estos algoritmos generan una solución óptima con un tiempo computacional muy alto, se pueden clasificar en tres grupos: búsqueda directa de árbol, programación dinámica, programación lineal y entera, entre los más destacados están los algoritmos de ramificación y acotación "*Branch & Bound*", del plano de corte o las técnicas de programación dinámica.

Estos métodos de búsqueda por ramificación y acotación, son ampliamente utilizados, y han resuelto satisfactoriamente algunos problemas combinatorios tan difíciles como el TSP. Su empleo se inició con los trabajos de Chirstofides 1970 y Held y Karp 1971. Para resolver el TSP. como lo indica Yepes, en [1].

En el método se considera un número fijo de m vehículos. Encuentra primero el costo mínimo alcanzable utilizando k vehículos, teniendo en cuenta la función del costo en la longitud de una ruta de vehículos a través de todos los vértices del subconjunto, luego encuentra el costo de todos los subconjuntos de vértices con m vehículos como lo describe L. B. Rocha Medina, et al, en[4].

En la literatura se encuentra a W. R. Golden, B.L.; Stewart,[35] en 1986 los cuales plantean la solución de problema de ruteamiento con ventanas de tiempo con programación dinámica, usando el método de Dantzing-Wolfe, la descomposición lagrangiana y la resolución directa clásica del modelo matemático. M. M. Koskosidis, et al. [36] desarrollaron un procedimiento donde el problema original se descompuso mediante heurísticas y luego se resuelven los otros más pequeños. Aunque en la actualidad se afirma que el método de Dantzing Wolfe es el que muestra mejores resultados.

H. W. J. M.Kolen, et al [37] presentan un algoritmo basado en la programación dinámica usando el método de ramificación y acotación consiguiendo soluciones de hasta 15 clientes de forma exacta en el VRPTW.

Algoritmos de solución aproximados

Cuando la cantidad de combinaciones de un problema es demasiado alta, el tiempo de proceso de un ordenador es demasiado alto, por ejemplo si tenemos el problema del agente viajero que debe visitar 25 puntos, entonces se tienen $25!/2$ combinaciones lo que

da como resultado un valor de 7.76×10^{22} que es un número grande de posibilidades para un número relativamente pequeño de puntos a visitar, además que el crecimiento de las combinaciones es de tipo exponencial, por esto en muchos de los problemas reales se vuelve imposible usar métodos exactos para encontrar la solución óptima incluso para los procesadores de alto rendimiento. Un ejemplo de la clasificación de este tipo de problemas combinatoriales y de solución no polinómica se muestra en la página 7 del libro "Técnicas metaheurísticas de optimización"[38]

Sin embargo se tiene que dar solución al problema sin importar su dificultad, en un tiempo razonable, así estos métodos exploran algunos espacios de soluciones que al parecer sean los que presenten mayor indicio de que exista una solución adecuada, lo que hace que el tiempo de búsqueda sea más bajo y práctico para un sistema computacional, generándose así una solución válida conocida como óptimo local o relativo. Estas técnicas se conocen como heurísticas y en otros casos como metaheurísticas.

Las heurísticas son métodos sencillos que proporcionan soluciones buenas a un problema dado mediante algoritmos específicos o constructivos que realizan una búsqueda local y las metaheurísticas hacen referencia a métodos generales, aplicables a amplios conjuntos de problemas, que emulan estrategias eficientes empleadas por la naturaleza y sirven para guiar el funcionamiento de las heurísticas.

Técnicas heurísticas

Las soluciones a los distintos problemas se inician a plantear desde los años sesentas y setentas, donde Fisher clasifica los métodos de solución en tres grupos, los de heurísticas simples basadas en búsqueda local, la segunda se basó en heurísticas aplicadas a la programación matemática para encontrar métodos casi exactos de resolución y la última, se basa en algoritmos exactos de optimización y en métodos basados en metaheurísticas como se indicaran más adelante.

Las técnicas heurísticas se definen en [1], como "procedimientos simples, a menudo basados en el sentido común, que se supone ofrecerán una buena solución (aunque no necesariamente la óptima) a problemas difíciles, de un modo fácil y rápido". Los procedimientos proporcionan soluciones de aceptable calidad mediante una exploración limitada del espacio de búsqueda. Clarke y Wright, propusieron el primer algoritmo que resultó efectivo para resolver el VRP en 1964. La mayoría de las heurísticas clásicas para resolver el VRP fueron desarrolladas entre 1960 y 1990. Estos métodos parten de rutas que contienen un único nodo para encontrar el mejor par (nodo, ruta) que representa la mejor intersección como lo indica L. B. Rocha Medina, et al, en[4].

Existen algunos elementos principales de estas así:

- a. **Heurísticas de construcción de soluciones factibles** basadas en añadir paulatinamente componentes individuales a la solución, hasta que se obtiene una opción viable. Uno de los más populares lo constituyen los algoritmos voraces “*greedy*”, que construyen paso a paso la solución buscando el máximo beneficio en cada etapa.
- b. **Heurísticas de descomposición** bajo el principio de “divide y vencerás”, fragmenta el problema en otros más pequeños de forma que al resolverlos todos se obtenga una solución para el problema global.
- c. **Heurísticas de reducción**, simplificando el problema tratando de distinguir alguna característica que presumiblemente deba poseer la solución óptima.
- d. **Heurísticas de manipulación del modelo** que trata de simplificar el esquema teórico para encontrar los valores de las variables de decisión con mayor facilidad, deduciendo, a partir de ella, la solución del problema original.
- e. **Heurísticas de búsqueda local** basados en la exploración del entorno de una solución. Este ámbito está formado por las opciones generadas por una alteración de la solución actual. Mediante estos movimientos se pasa iterativamente de una solución a otra mientras no se cumpla un determinado criterio de terminación. Un procedimiento de búsqueda local queda completamente determinado al especificar un movimiento y el criterio de selección de una solución dentro del entorno.

Este tipo de problemas inicialmente se plantean soluciones a través de los modelamientos matemáticos y la búsqueda de soluciones exactas por medio de algoritmos que realicen una exploración de las rutas posibles dividiendo el problema en diferentes partes o tomando algunas de las posibles combinaciones de la manera más conveniente como por ejemplo seguir al más próximo, intercambios de arcos, entre otros.

Como algoritmos de construcción o heurísticas se puede generar soluciones tanto secuenciales como en paralelo con la diferencia que los secuenciales construyen una ruta a la vez y los paralelos de forma simultánea con el criterio de averiguar que cliente es el que se debe insertar en cada iteración y a que ruta.

J. R. Baker y E.K Schaffer [39] establecieron uno de los primeros métodos de construcción de rutas, proponen una variante a la heurística clásica propuesta por Clarke y Wright [40] para el VRP. El algoritmo comienza con una ruta para cada uno de los clientes. En cada iteración se calcula qué par de itinerarios pueden agruparse de modo que la economía conseguida sea máxima, es decir cuánto ahorra ir de un nodo a otro o no ir.

M.M Koskosidis, et al en [36] diseñó un conjunto de heurísticas de construcción de rutas, su método comienza con un trayecto que desde el almacén se dirige a un nodo seleccionado previamente luego regresa a su origen. Se elige al cliente más alejado de la base o bien a aquel otro cuyo límite horario de inicio del servicio sea más temprano. Posteriormente se evalúan todas las inserciones posibles de los nodos que aún no han

sido visitados. Se opta por aquella que disminuya una suma ponderada de la distancia y el tiempo requerido. Por último, se selecciona al cliente utilizando una generalización de la expresión del máximo ahorro de Clarke y Wright [40]. Una nueva ruta comienza cuando es imposible intercalar más nodos.

Sin embargo, y pese a ser el algoritmo de M.M Koskosidis, et al [36] rápido y aportar buenas soluciones en numerosas ocasiones, hoy no es competitivo frente a otros procedimientos metaheurísticos posteriores.

J.M Potvin y J.Y Rousseau [41] adoptan un procedimiento de construcción de rutas en paralelo que mejora el algoritmo de Solomon cuando los clientes se distribuyen de forma aleatoria por el espacio. Ello se explica por la miopía de los enfoques secuenciales. En algunos casos, los últimos clientes no asignados quedan diseminados, de modo que las últimas rutas son frecuentemente de peor calidad. Sin embargo, el algoritmo de Solomon funciona mejor que el propuesto cuando los clientes se encuentran agrupados por zonas.

U. Antes y J. Derigs [42] también se aproximan al problema mediante la idea, ya clásica, de la inserción. Cada cliente no asignado solicita y recibe de cada ruta un precio para incorporarse definida de forma similar a la métrica de M.M Koskosidis, et al [36]. Los resultados son parejos a los obtenidos en [41].

Posteriormente, J.F Kontoravdis y G.A Bard [43] emplearon métodos paralelos de construcción de rutas inmersos en procesos de mejora, como se tendrá ocasión de comprobar en el siguiente apartado.

Como mejoramiento de las rutas se presentan heurísticas que buscan el paso de una solución viable a otra que genere menor costo en la función objetivo con varias iteraciones, aunque siempre se obtendrá un óptimo local. Por lo que se conocen como heurísticas de búsqueda local o por el método numérico para la solución que generalmente se usa gradiente descendente.

El algoritmo que genera buenos resultados en la búsqueda local es el denominado K-intercambio o *K-OPT*, en el que generalmente se intercambian arcos de la solución actual durante varias iteraciones tanto en una sola ruta o en varias.

Otra forma de generar intercambios es la denominada “Movimiento 1-swap” basada en intercambiar dos nodos de la misma ruta. A diferencia del 2-opt, se conserva la orientación de los segmentos que no se han visto alterados, lo cual favorece la resolución de problemas con restricciones de tiempo.

Cuando se genera el salto de un nodo a otra posición dentro de su misma ruta, a este movimiento se lo denomina 1-relocate.

O. Bräysy [44] en el año 2001 propone el movimiento IOPT generalizando el movimiento Or-opt para el caso de una sola ruta. Se trata de cambiar un tramo dentro de una ruta, aunque en este caso, con un cambio en el sentido de avance. Este operador tiene éxito cuando los dos nodos se encuentran próximos.

J. Christofides y N. Beasley [45] hablan de movimientos entre dos rutas, es decir se intercambian nodos de rutas distintas, a esto lo denominan Movimiento *I-Exchange*.

S. R. Balseiro, et al[20] muestran como Potvin y Rousseau, generan intercambios de *2-OPT* con la condición de que existan varias rutas así los primeros clientes de la primera ruta se unen a los últimos clientes de la segunda y viceversa. De esta forma se conserva la orientación en las trayectorias, pues la introducción de clientes con restricciones temporales, así las rutas que generan mayores tiempos se enlazan con aquellos que no los generan. Ello favorece que las soluciones resultantes sean factibles. También se propuso un tipo de intercambio que considera la cadena de uno, dos o tres clientes consecutivos de una solución y la mueve a otro lugar dentro de la misma ruta o incluso de otra.

Otra forma de generar intercambios es por cruce y es una generalización de los métodos *2-opt* y *Or-opt* que, además, mantiene la orientación de las rutas, lo cual favorece la solución de problemas con restricción temporal.

El procedimiento de las transferencias cíclicas según H.N Thompson y P.M Psaraftis [46], consiste en trasladar simultáneamente k nodos entre un conjunto r de rutas.

Para la generación de mejoras en las soluciones se trata de combinar las estrategias anteriores o usar varias al tiempo, aunque también se puede realizar otros intercambios como los siguientes:

El salto secuencial “shift-sequence” propuesto por T. Schulze y J. Fahle, [47] trata de eliminar un nodo de una ruta e incorporarlo a una nueva posición de cualquier otra ruta, ensayando todas las opciones. Si alguna de ellas es factible, se permite el movimiento y se continúa hasta que sea posible.

La operación de Reducción de rutas selecciona aquel itinerario con un mejor número de nodos. Seguidamente, se desconectan estos nodos y se asignan, uno a uno, a aquellas rutas y lugares que maximizan la función objetivo, incluyéndose el caso del inicio de un nuevo recorrido.

También se puede generar un Intercambio de vehículos que realizan dos rutas distintas, o sustituir vehículos que realizan una ruta por otro que no está en esa ruta pero que no se ha tomado en otra antes.

La combinación de varias heurísticas se denomina heurísticas mixtas, J.F Kontoravdis y G.A Bard [43] idearon una heurística GRASP (procedimiento ávido de búsqueda adaptativa aleatoria) que promueve rutas iniciales en paralelo que se mejoran con procedimientos de búsqueda local.

Técnicas metaheurísticas

Según J.P Osman y I.H Kelly [48] definen como “Los procedimientos metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son ni efectivos ni eficientes. Los metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y la mecánica estadística”. Principalmente estas técnicas permiten el uso de métodos y características de la naturaleza para encontrar algún tipo de soluciones confiables y rápidas, aunque no sean las óptimas; sin embargo el objetivo de estas técnicas es aportar a la intensificación de la búsqueda de mejores soluciones incluso seleccionando algunas que sean peores pero que puedan en algún momento ayudar a salir de soluciones óptimas locales.

Algunas de las metaheurísticas más conocidas se mencionan a continuación.

a. Búsqueda secuencial por entornos

La idea central de las metaheurísticas basadas en las búsquedas por entornos se fundamenta en la degradación estratégica de las opciones que mejoran las de su vecindario para alcanzar un nuevo óptimo relativo usando operaciones que permitan el paso de una solución a otra en su entorno para mejorar la función objetivo hasta encontrar un óptimo local bueno o un óptimo global después de un gran número de iteraciones.

Esta metaheurística fue propuesta por Hansen y Mladenovic según lo describe C. Martínez en su tesis [49]. Esta metaheurística permite resolver problemas de optimización por medio de la búsqueda de soluciones competitivas mediante el cambio sistemático de vecindarios (también conocidos como entornos). VNS explora vecindarios distantes de la solución actual de manera incremental, y salta desde el mismo a uno nuevo si se ha obtenido una mejora. La idea detrás de VNS es el cambio dinámico de vecindarios dentro de un algoritmo de búsqueda local.

Los principios básicos en los que se basa esta metaheurística son:

- Un óptimo local con una determinada estructura de vecindario, no necesariamente lo es con otra vecindad.
- Un óptimo global es un óptimo local con todas las estructuras de vecindarios.

- Para muchos problemas de optimización, los óptimos locales están relativamente cerca (entre ellos).
- En base a los dos primeros hechos, se desprende que es necesario incorporar varias estructuras (parametrizadas) de vecindarios en los algoritmos basados en VNS, para obtener soluciones de calidad. Por esto, en base a la información provista por los óptimos locales, es posible alcanzar el óptimo global. Dado que entre un óptimo local y el óptimo global, los valores que toman ciertas variables de decisión (del problema) son los mismos, mediante un estudio adecuado, es posible dirigir el proceso de búsqueda hacia la solución óptima.

b. Algoritmos de destrucción y reconstrucción

El principio de destrucción y reconstrucción de soluciones es una estrategia que permite obtener opciones de mayor calidad en los problemas combinatorios. Se trata, en primer lugar, de romper una parte significativa de una propuesta inicial para después reconstruirla de modo que la alternativa tenga opciones de mejorar la función objetivo de la solución inicial. La reconstrucción de soluciones supone un operador o movimiento complejo que define un entorno de la solución inicial y un óptimo relativo. Son aplicables, por tanto, cualesquiera de las metaheurísticas diseñadas para eludir los mínimos locales y, a su vez, es combinable con otros movimientos más simples para ampliar el entorno y las posibilidades de encontrar mejores soluciones.

El funcionamiento de esta técnica se basa en tomar un radio determinado y los vecinos al punto que se desea visitar, en una primera fase se rompen aquellos enlaces que conectan los nodos tomando como semilla la solución encontrada hasta el momento, para luego reconstruir una solución que mejore, si es posible, a la inicial.

La calidad del método radica en definir un procedimiento de reparación de la solución destruida que permita obtener, con una probabilidad razonable, una opción que mejore la función objetivo. P. Shaw [50], propuso llamar a esta técnica como búsqueda en entornos amplios *Large Neighborhood Search* (LNS) de modo que tras romper una parte de la solución, se recompone empleando técnicas de programación condicionadas para optimizar la reconstrucción.

c. Colonia de hormigas

Los autores V. Coloni, et al [51] sugirieron la idea de imitar el comportamiento de los insectos para encontrar soluciones a los problemas de optimización combinatoria. El principio de la metaheurística denominada como "*Ant System Optimization, ACO*" se basa en el comportamiento colectivo de las hormigas en la búsqueda de alimentos para su subsistencia, que son capaces de encontrar el camino más corto entre una fuente de comida y su hormiguero.

Primero las hormigas exploran el entorno de su hormiguero de forma aleatoria. Tan pronto como un individuo encuentra una fuente de comida, evalúa su cantidad y calidad y transporta un poco al hormiguero. Durante el regreso, la hormiga deja por el camino una señal depositando una sustancia denominada feromona, para que las demás puedan seguirla. Después de un tiempo, el camino hacia el alimento se indicará por un rastro que crece con el número de hormigas que pasen por él, y que va desapareciendo en caso contrario. El resultado final es la optimización del trabajo de todo el hormiguero en su búsqueda de comida.

Según los autores G. Gambardella, et al.[52], resuelven el problema de ruteamiento con ventanas de tiempo empleando la semejanza del comportamiento las hormigas usando dos hormigueros artificiales que colaboraban jerárquicamente mediante el intercambio de información en la actualización de las feromonas. El primero intenta reducir el número de vehículos, mientras que el segundo lo hace con la distancia total recorrida. El efecto de la actualización local de la feromona es el cambio dinámico de la atracción de los arcos, cada vez que una hormiga utiliza un trayecto, la cantidad de feromona asociada decrece y se vuelve menos atractivo. Por otra parte, el reinicio global intensifica la búsqueda en las inmediaciones de la mejor solución encontrada y se puede intensificar la búsqueda con intercambios o cruces.

d. Algoritmos genéticos

Básicamente, los algoritmos genéticos, simulan el proceso de evolución de las especies que se reproducen sexualmente. De manera muy general, se puede decir que en la evolución de los seres vivos, el problema al que cada individuo se enfrenta diariamente es el de la supervivencia. Para ello cuenta, entre otras, con las habilidades innatas provistas en su material genético. A nivel de los genes, el problema consiste en la búsqueda de aquellas adaptaciones beneficiosas en un medio hostil y cambiante. Debido en parte a la selección natural, cada especie gana cierta “información” que es incorporada a sus cromosomas.

Durante la reproducción sexual, un nuevo individuo, diferente de sus padres, se genera a través de la acción de dos mecanismos fundamentales: El primero es el *cruzamiento*, que combina parte del patrimonio genético de cada progenitor para elaborar el del nuevo individuo; el segundo es la *mutación*, que supone una modificación espontánea de esta información genética. La descendencia será diferente de los progenitores, pero mantendrá parte de sus características. Si los hijos heredan buenos atributos de sus padres, su probabilidad de supervivencia será mayor que aquellos otros que no las tengan. De este modo, los mejores tendrán altas probabilidades de reproducirse y diseminar su información genética a sus descendientes.

J.H Holland [53] estableció por primera vez una metaheurística basada en la analogía genética. Un individuo se puede asociar a una solución factible del problema, de modo que se pueda codificar en forma de un vector binario. Entonces un operador de cruzamiento intercambia cadenas de los padres para producir un hijo. La mutación se configura como un operador secundario que cambia, con una probabilidad pequeña, algunos elementos del vector hijo. La aptitud del nuevo vector creado se evalúa de acuerdo con una función objetivo.

De forma algorítmica se tiene:

1. Generar una población de vectores (individuos).
2. Mientras no se encuentre un criterio de parada:
 - Seleccionar un conjunto de vectores padre, que serán reemplazados de la población.
 - Emparejar aleatoriamente a los progenitores y cruzarlos para obtener unos vectores hijo.
 - Aplicar una mutación a cada descendiente.
 - Evaluar a los hijos.
 - Introducir a los hijos en la población.
 - Eliminar a aquellos individuos menos eficaces.

Normalmente este proceso finaliza después de un número determinado de generaciones o cuando la población ya no puede mejorar. La selección de los padres se elige probabilísticamente hacia los individuos más aptos. Al igual que ocurre con en la Naturaleza, los sujetos con mayor aptitud diseminan sus características en toda la población.

3. METODOLOGÍA DE SOLUCION PARA EL PROBLEMA TDVRP

En este capítulo se hace inicialmente una descripción del proceso de búsqueda de soluciones al CVRP por medio del Algoritmo genético modificado de Chu-Beasley, cabe notar que un punto importante en la lógica del algoritmo genético es la generación de la población inicial, la cual está formada por individuos tomados aleatoriamente y otros encontrados con métodos de solución heurísticos y metaheurísticos, los cuales se describirán en una segunda instancia.

Por otra parte, con respecto a la dependencia del tiempo en el problema, la metodología indica que se debe ajustar en la función objetivo del modelo por medio de la matriz de costos del problema y el factor de sobrecosto dado en función de la congestión vehicular, por lo que en principio se resuelve el CVRP y al aplicar su solución a la realidad del problema, junto con los requerimientos de la empresa COLACTEOS, se corre el algoritmo como un CVRP con la matriz de costos afectada por la congestión en los diferentes horarios de salida de los vehículos de la empresa. La matriz de costos mencionada se detalla en capítulos siguientes.

3.1 ALGORITMO GENÉTICO MODIFICADO EN TDVRP

El TDVRP corresponde al tipo NP-HARD, por lo que se requiere desarrollar e implementar una metodología adecuada para su solución, en este caso se opta por el uso del Algoritmo Genético de Chu-Beasley como principal herramienta metaheurística para establecer las rutas de los vehículos dependientes de tiempo, teniendo en cuenta como función objetivo la minimización de los costos (Ver *restricción 2-19*) dados por la matriz de distancias entre los clientes más el factor de sobrecosto por la restricción del tiempo u horario de salida de los vehículos para todas las visitas desde el depósito y con retorno a él en cada ruta.

Los elementos como matriz de costos se toman inicialmente con casos de prueba para distinto número de clientes y para el caso real se encuentra la matriz de costos en la ciudad de San Juan de Pasto como se indica en el capítulo 4.

De igual manera en el capítulo 5 se muestra el cálculo del factor de sobrecosto por condiciones temporales.

En este capítulo se hace referencia en el algoritmo a las distintas restricciones del modelo, tales como la capacidad del vehículo, la demanda de las ciudades, el tipo de variables y sus valores.

A continuación se enumera una serie de pasos lógicos del proceso algorítmico:

Paso 1. Generación de la población inicial de forma aleatoria (200 individuos).

Paso 2. Partición de vectores de acuerdo a la demanda para generar cada una de las rutas.

Paso 3. Cálculo de la función objetivo. Según (2-19), matriz de costos y sobrecostos por congestión vehicular.

NOTA: En el diagrama de flujo de la figura 3-1 se indica como punto de partida la configuración de parámetros iniciales, los cuales consisten en determinar el número de clientes, la ubicación geográfica, costos de distancia y sobrecostos por horario de salida de acuerdo a la congestión vehicular dada por un análisis estadístico (ver capítulo 4 y 5).

Paso 4. Realizar selección de padres por medio de técnica de ruleta.

Paso 5. Realizar recombinación de padres por medio de intercambio de cromosomas.

Paso 6. Partición de vectores hijos, de acuerdo a la demanda para generar cada una de las rutas y generar intercambio 2 OPT en cada ruta.

Paso 7. Hacer mutación si corresponde y calcular valor de la función objetivo. Según (2-19) y matriz de costos para individuos hijos.

Paso 8. Decidir que individuo ingresa a la población y que individuo se deshecha de acuerdo a un criterio de aspiración.

Paso 9. Verificar cumplimiento de iteraciones. Si se cumple, ir al paso 10, en caso contrario ir a paso 4.

Paso 10. Verificar si ya se realizó la modificación de población. Si es así, ir al paso 14, en caso contrario ir al paso 11.

Paso 11. Marcar modificación de la población

Paso 12. Reiniciar iteraciones y reemplazar a los peores individuos de la población por nuevos individuos (20) a través de métodos heurísticos y metaheurísticos listados a continuación:

- a. Heurística de intercambio 2 OPT.
- b. Heurística de inserción del más barato.
- c. Heurística del vecino más cercano
- d. Heurística del ahorro
- e. Heurística del barrido.
- f. Colonia de hormigas.
- g. Colonia de hormigas con intensificación de búsqueda.

NOTA: Los individuos generados, contienen el orden de las visitas, la partición de rutas y el valor de su función objetivo.

Paso 13. Verificar cumplimiento de iteraciones. Si cumple, ir a paso 14, en caso contrario ir a paso 4.

Paso 14. Se elige la mejor respuesta de la población y se hace una búsqueda en entornos variables.

Paso 15. Realizar intercambio 2 OPT en cada ruta y dar la respuesta final.

Como se observa en el listado de pasos, existe una población para el algoritmo genético dada inicialmente de forma aleatoria y mejorada por un conjunto de iteraciones propias

de la metodología de los algoritmos genéticos, las cuales hacen parte del conjunto de individuos iniciales que se adicionan con otros individuos generados por distintos métodos (Paso 12) , conformando la población inicial, a la que se genera un conjunto de iteraciones según la metodología del algoritmo genético de Chu – Beasley.
Una representación general en un diagrama de flujo se indica a continuación:

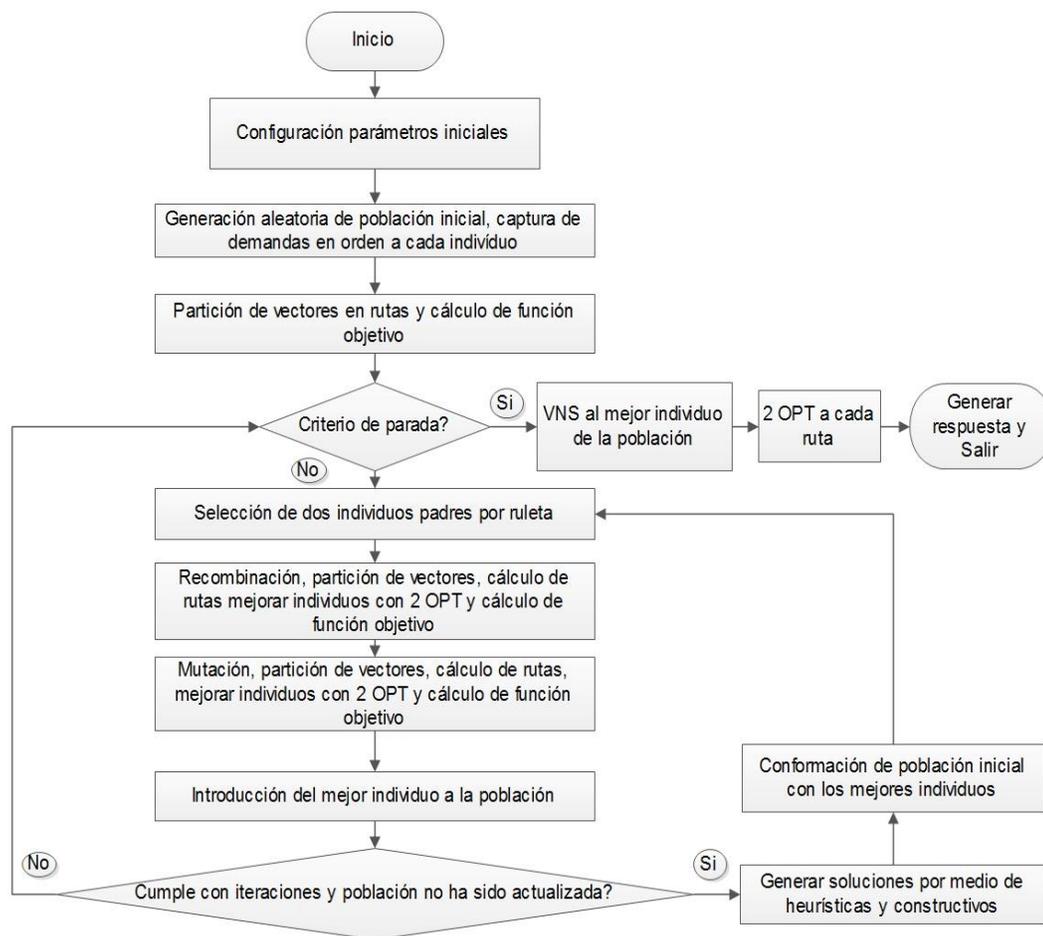


Figura 3-1 Diagrama de flujo algoritmo genético

A continuación se realiza una descripción más detallada de la solución del problema.

3.1.1 Representación del cromosoma:

Los individuos de la población están formados por valores enteros desde el número 2 hasta el número de clientes o ciudades a visitar en un vector, es así como para 10 clientes

en los que se incluye el depósito, el vector estará conformado por 9 números enteros entre el 2 y el 10 que corresponden a los diferentes clientes. Se excluye el valor numérico 1, puesto que este corresponde a la identificación del depósito. Por otra parte a cada individuo de la población le corresponde un vector de vehículos de la misma dimensión del vector clientes con el fin de dar libertad al número de rutas posibles. En cada posición del vector se indica el número de clientes a visitar de forma consecutiva. Así, para 9 clientes y el depósito se tiene:

7	2	5	6	3	8	4	9	10
---	---	---	---	---	---	---	---	----

Figura 3-2. Vector de secuencia de clientes

4	2	3	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

Figura 3-3. Vector de vehículos del individuo.

En los vectores se muestra como al individuo se le asignan 3 rutas, de forma que el vehículo 1, visita a las 4 ciudades iniciales del vector de clientes, es decir que sale del depósito, se dirige a la ciudad 7, luego a la 2, continúa a la 5 y luego la 6 y finaliza en el depósito.

El vehículo 2, atiende a las ciudades 3 y 8.

El vehículo 3, atiende a las ciudades 4, 9 y 10.

La dimensión del vector de vehículos es igual a la del vector clientes y el número de rutas suficientes para los clientes es de 3, los demás elementos del vector de vehículos tienen el valor de 0, el cual se usa como criterio de parada para el cálculo de la función objetivo.

Se nota además que la suma del vector de vehículos debe ser equivalente al número de ciudades menos uno. ($4+2+3=9$).

A continuación se muestra un individuo de la población con 9 clientes más el depósito:

El corte de las rutas en los individuos de la población se realiza de acuerdo a la demanda correspondiente a cada cliente en el orden del individuo.

Usando vehículos con capacidad de 50 toneladas y el vector de demanda de los 9 clientes:

12	8	30	4	16	2	3	7	5
----	---	----	---	----	---	---	---	---

Figura 3-4. Demanda de clientes

Se tiene que la demanda total es de 87 toneladas, lo cual estimaría tener 2 vehículos para cumplir con la demanda solicitada, sin embargo se deja libertad para tener 9 rutas máximo.

Los vehículos son asignados al recorrido desde la posición inicial hasta que se termine la capacidad del mismo.

7	2	5	4	6	9	3	8	10
---	---	---	---	---	---	---	---	----

Figura 3-5. Vector Clientes

2	12	4	30	16	7	8	3	5
---	----	---	----	----	---	---	---	---

Figura 3-6. Demanda en orden de los clientes

El vehículo 1, con capacidad de 50 unidades puede visitar a los clientes 7,2,5,4 puesto que su demanda suma 48 unidades.

El vehículo 2 tendra que visitar a los clientes 6,9,3,8 y 10 puesto que la demanda de estos es de 39 unidades.

A cada individuo de la población se le hace corresponder el valor de la función objetivo, de acuerdo con los recorridos de cada ruta con salida y llegada al depósito, la cual es calculada de acuerdo a la matriz de costos y el orden en que se realiza las visitas.

Paso 1: Tomar el vector de clientes, $C[ii,:]$, el vector de vehículos $V[ii,:]$ del individuo ii de la población, junto con la matriz de distancias o costos de los clientes. $D[]_{n \times n}$. (n , indica el número de clientes incluido el depósito).

Paso 2: Recorrer el vector de vehículos $V[i,:]$ desde la posición i , hasta la posición en que exista un valor de cero. (Mientras $V[i,:] \neq 0$)

Paso 3: Tomar el valor de clientes a recorrer como lo indique el vector de vehículos en la posición i : ($cl=V(i)$)

Paso 4: Desde 1 hasta $cl-1$: Tomar el valor de costo en $D(C(i),C(i+1))$. y acumularlo en *CostoRuta*.

Paso 5: Sumar el *CostoRuta* junto con el costo de ir desde el deposito al cliente inicial de la ruta y del cliente final de la ruta al depósito. ($D(1,C(1)) + D(cl,1)$).

Paso 6: Función objetivo = Funcion objetivo + CostoRuta.

Paso 7: Hacer $i=i+1$. Si $V(i)=0$ Terminar, en caso contrario ir al Paso 2.

Para generar mayor claridad en este procedimiento se presenta un ejemplo con 9 ciudades, con el vector de demandas anterior y capacidad de 50 toneladas por los vehículos y una matriz de distancias D :

Tabla 3-1 Matriz de distancias para 10 ciudades.

Ciudad	1	2	3	4	5	6	7	8	9	10
1	0	8	3	4	16	2	3	7	5	12
2	8	0	4	6	3	8	9	12	2	3
3	3	4	0	3	5	6	8	12	4	5
4	4	6	3	0	2	5	6	8	9	8

5	16	3	5	2	0	1	4	5	7	6
6	2	8	6	5	1	0	2	4	8	23
7	3	9	8	6	4	2	0	13	21	9
8	7	12	12	8	5	4	13	0	4	13
9	5	2	4	9	7	8	21	4	0	6
10	12	3	5	8	6	23	9	13	6	0

3	5	10	2	6	9	4	7	8
8	4	5	12	16	7	30	2	3

Figura 3-7. Vector clientes y Vector de demandas correspondientes a cada cliente

En el vector se nota la demanda de la ciudad respectiva.

Para el individuo se asigna al primer vehículo que recorre desde el depósito o cliente 1, pasando por los clientes 3, 5, 10, 2 y 6 y regresa al depósito. Con una carga de $8 + 4 + 5 + 12 + 16 = 45$. La ruta no puede tomar al cliente 9, puesto que se excede la capacidad del vehículo correspondiente a 50 Toneladas, por lo que se toma una siguiente ruta con el vehículo 2 que recorre desde el depósito a los clientes 9,4,7 y 8.

El vector de vehículos se representa así:

5	4	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

Figura 3-8. Vector de vehículos del individuo.

Con estas rutas se determina el valor de la función objetivo a partir de la suma de distancias desde el depósito (1) más el recorrido de los diferentes clientes y el regreso al depósito así:

Función objetivo:

$$\text{Ruta 1} = 3 + 5 + 6 + 3 + 8 + 2 = 27$$

$$\text{Ruta 2} = 5 + 9 + 6 + 13 + 7 = 40$$

Dando un total de costo de 67.

3.1.2 Ajuste de parámetros:

El algoritmo implementado en el *software matlab*, consiste principalmente en seguir la metodología de Chu-Beasley, es decir que en principio se define una población inicial de individuos que sean posibles soluciones al problema, generando varios vectores con secuencias de visitas a los clientes, cada vector de una población es entonces una solución que se genera inicialmente de forma aleatoria, cada individuo de la población tiene

asociado un vector de demanda correspondiente a los clientes a visitar y el número de vehículos que se requiera según la carga, como se indicó en la representación del cromosoma.

El tamaño de la población, el número de generaciones, si se requiere de uno o dos puntos de recombinación, al igual que si se toma uno o dos genes para la mutación. Estos parámetros se eligen de acuerdo al rendimiento que tenga el algoritmo por medio de ensayos en diferentes corridas.

A continuación se indica un resumen de algunas corridas del algoritmo con algunos valores en la función objetivo para diferentes parámetros. La población para todos los casos se toma con 20 individuos generados por heurísticas y otros de forma aleatoria.

Tabla 3-2. Pruebas para mutación de un gen y un punto de recombinación

Corridas	Población Aleatoria	Generaciones	Tiempo computacional	F.O CVRP	% error
1	30	500	55.7231	999.7288	6.6946
2	60	500	49.9994	1.015,9	8.4225
3	100	500	50.6704	1.001,6	6.8962
4	200	500	51.0389	990.6774	5.7286
5	400	500	51.0083	998.0014	6.5103
6	30	1000	75.3727	1.043,4	11.3514
7	60	1000	77.8608	977.4704	4.3191
8	100	1000	77.9049	1.043,3	11.3418
9	200	1000	79.0177	962.8023	2.7537
10	400	1000	83.5849	1.121,1	19.6455
11	30	3000	209.6615	1.064,5	13.6031
12	60	3000	179.3649	967.2272	3.2260
13	100	3000	166.2124	1.020,3	8.8886
14	200	3000	165.9097	965.3680	3.0275
15	400	3000	204.4481	969.2988	3.4470

La tabla 3-2, muestra como mejor resultado el usar una población de 200 individuos generados de forma aleatoria y con 1000 iteraciones, para un punto de recombinación y un gen mutado.

Tabla 3-3. Pruebas para mutación de dos genes y un punto de recombinación

Corridas	Población Aleatoria	Generaciones	Tiempo computacional	F.O CVRP	% error
16	30	500	51.2072	1.079	15.1600
17	60	500	56.6386	994.52	6.1388
18	100	500	53.4527	1.098,2	17.2033
19	200	500	55.4507	1.005,1	7.2673
20	400	500	103.4951	1.113,2	18.8096
21	30	1000	135.6526	1.068,4	14.0278
22	60	1000	140.7480	1.002,2	6.9612
23	100	1000	97.6104	989.9966	5.6560
24	200	1000	161.5061	973.0078	3.8429
25	400	1000	169.3873	1.024,6	9.3448

26	30	3000	219.3223	1.021,4	9.0119
27	60	3000	157.4341	1.099,4	17.3302
28	100	3000	154.4921	973.8646	3.9343
29	200	3000	203.4901	1.017,6	8.5995
30	400	3000	183.9519	1.010,9	7.8871

En la tabla 3-3, se repite el mejor valor usando 1000 generaciones y 200 individuos, aunque empeora el tiempo de ejecución del programa aproximadamente en el doble con respecto al uso de un solo gen de mutación. Lo que lleva a intuir que es mejor usar un solo gen para el proceso de mutación.

Tabla 3-4. Pruebas para mutación de un gen y dos puntos de recombinación.

Corridas	Población Aleatoria	Generaciones	Tiempo computacional	F.0 CVRP	% error
31	30	500	51.7154	1.025,5	9.4420
32	200	500	53.9055	996.0502	6.3021
33	400	500	54.5501	992.3383	5.9059
34	30	1000	68.2602	976.3355	4.1980
35	200	1000	83.9166	971.8586	3.7202
36	400	1000	85.1878	1.043,9	11.4038

La tabla 3-4, indica una mejoría al usar dos puntos de recombinación con la población de 200 individuos aleatorios y 1000 generaciones. El tiempo de respuesta es comparable con el caso de un punto de recombinación y un gen mutado.

Tabla 3-5. Pruebas para mutación de dos genes y dos puntos de recombinación

Corridas	Población Aleatoria	Generaciones	Tiempo computacional	F.0 CVRP	% error
37	30	500	57.7675	992.0072	5.8706
38	200	500	53.3464	967.6105	3.2669
39	400	500	55.2786	1.017,5	8.5891
40	30	1000	72.7190	997.3446	6.4402
41	200	1000	81.6386	949.8775	1.3743
42	400	1000	85.8265	977.3662	4.3080

La tabla 3-5, muestra un valor de mejores características para las 1000 iteraciones y 200 individuos, con un tiempo computacional semejante a los mejores casos anteriores.

Finalmente, se encuentra que un buen valor para los parámetros del algoritmo genético están dados en 200 individuos, 1000 iteraciones, dos puntos de recombinación y dos genes para el proceso de mutación.

Por otra parte se puede observar graficamente el comportamiento de la función objetivo de acuerdo a las iteraciones del algoritmo, así:

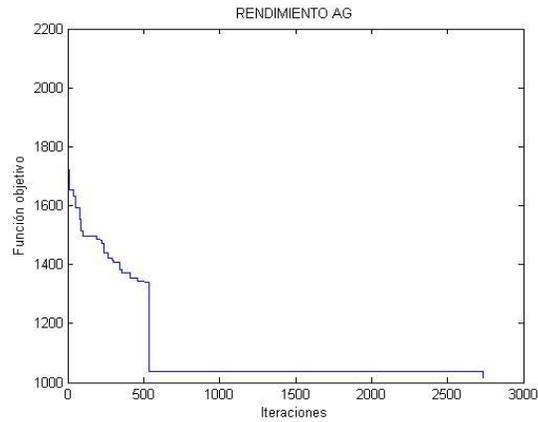


Figura 3-9. Rendimiento del algoritmo para población de 30 individuos

En la figura 3-9 se toma un valor de 30 individuos en la población y se nota como desde la iteración 500 en adelante el valor de la función objetivo permanece constante, es decir que no existe mejoramiento.

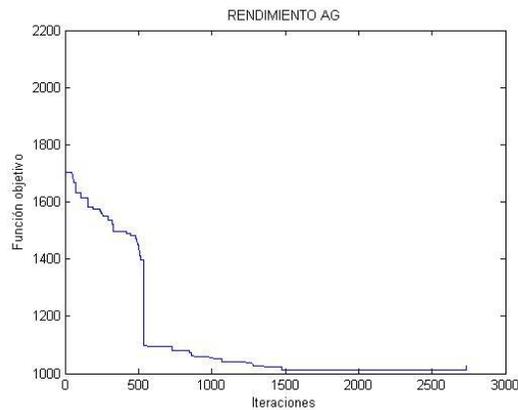


Figura 3-10. Rendimiento del algoritmo para población de 200 individuos

En la figura 3-10 se nota como al aumentar el número de individuos, la función objetivo continúa mejorando con las iteraciones, sin embargo cuando el número de individuos crece, el algoritmo se hace más lento para generar la respuesta y el mejoramiento es similar al de 200 individuos, como se observa en las tablas 3-2 a 3-5.

3.1.3 Selección:

Continuando con el proceso, los individuos de la población, están formados por los vectores de un numero de cromosomas enteros que corresponden a la permutación aleatoria del número de ciudades menos el deposito el cual corresponde al valor de 1, así: si se tiene un numero definido de 10 clientes, el vector solución estaría formado por la permutación de los números enteros desde el valor de 2 al número 10.

[2	8	3	4]	[6	9	10	7	5]
4	5	0	0	0	0	0	0	0
76								

Figura 3-11. Individuo de la población

En este paso luego de tener una matriz de población con los individuos, su vector de vehículos correspondiente a las rutas y el valor de la función objetivo, se pasa a realizar una selección aleatoria de dos individuos de la población que se denominan padres, que posteriormente se combinarán siguiendo las etapas del algoritmo genético.

La selección de dichos padres se realiza usando la técnica de la ruleta, la cual genera una probabilidad de ser elegido el individuo de la población, de acuerdo al valor de su función objetivo, es decir que los individuos con menor función objetivo, tienen mayor probabilidad de ser elegidos para la etapa de cruce.

3.1.4 Cruce de individuos:

La posibilidad de combinación, se hace seleccionando aleatoriamente uno o dos puntos de cruzamiento del vector en cada padre, por ejemplo:

Padre1:

3	5	10	2	6	9	4	7	8
---	---	----	---	---	---	---	---	---

Figura 3-12. Vector Padre 1

Si en el padre 1 se tiene el vector de 9 casillas se toma aleatoriamente una posición para realizar el cruce, por ejemplo con un punto de cruce si se selecciona aleatoriamente la posición 3.

Padre 2:

2	8	3	5	7	9	10	4	6
---	---	---	---	---	---	----	---	---

Figura 3-13. Vector Padre 2

Se realiza lo mismo con el padre 2, para este caso se puede suponer un valor aleatorio de 5 como punto de cruce.

Luego de tener las posiciones de cruce, se procede a intercambiar los cromosomas entre los padres, tomando inicialmente el punto de cruce del padre 1 y generándose 2 hijos así:

3	5	10	2	6	9	4	7	8
---	---	----	---	---	---	---	---	---

	2	8	3	5	7	9	10	4	6
Hijo 1:									
	3	5	10	5	7	9	10	4	6
Hijo 2:									
	2	8	3	2	6	9	4	7	8

Figura 3-14. Vectores Hijos por recombinación

Normalmente se encuentra que los hijos presentan 1 o más cromosomas repetidos, así si observamos el hijo1, nos damos cuenta que el valor de 5 y 10 están repetidos en el vector, por lo que se deben retirar las dos copias, reemplazándolos por los valores faltantes, en este caso los números 2 y 8 que se encuentran repetidos en el vector hijo2. Los cuales se buscan en el vector y se intercambian de forma ordenada, es decir se recorre el vector hijo1 y el primer término repetido se intercambia con el primer término repetido del vector hijo2, luego el segundo etc. Quedando los vectores hijos de la siguiente manera:

Hijo 1:									
	3	2	8	5	7	9	10	4	6
Hijo 2:									
	5	10	3	2	6	9	4	7	8

Figura 3-15. Vectores Hijos

Se realiza el mismo procedimiento con la posición de cruce aleatoria del padre 2. A cada vector Hijo 1 y Hijo2, se realiza la asignación de vehículos y cálculo de función objetivo como se indicó en la representación del cromosoma anteriormente. Posteriormente se debe realizar la eliminación de uno o dos individuos de la población, ya sean hijos o padres. Para ello es importante notar que como objetivo se requiere minimizar el costo de distancias, la cual está definida como una matriz que contiene el valor de cada cliente, incluido el depósito, es decir que si por ejemplo existen 50 clientes, entonces se tiene una matriz simétrica de 51x51.

3.1.5 Mutación:

En el algoritmo genético se realiza este proceso durante un numero grande de iteraciones, además se tiene en cuenta una probabilidad de mutación en los hijos que consiste en que a un 0.5% de los hijos después del cruce se realice uno o dos intercambios entre dos cromosomas del mismo hijo de forma aleatoria y se evalúa en la función objetivo, si esta mejora se deja al hijo mutado, de lo contrario se desecha al hijo mutado, aunque en algunos casos cuando la población genera un óptimo local y se encuentra que todos los individuos toman valores iguales en la función objetivo,

entonces se deja al hijo mutado aunque empeore el valor de la función objetivo pero puede refrescar la población permitiendo salir del óptimo local hacia otro óptimo. El valor de probabilidad se selecciona por ensayo y error ajustando en varias corridas desde un valor de 0.05% al 15%, notándose un resultado favorable en el valor de 0.5% como se indica en la figura 3-16. El algoritmo se corre para una población de 200 individuos con 500 iteraciones y 30 valores de probabilidad de mutación de uno y dos genes, con dos puntos de recombinación.

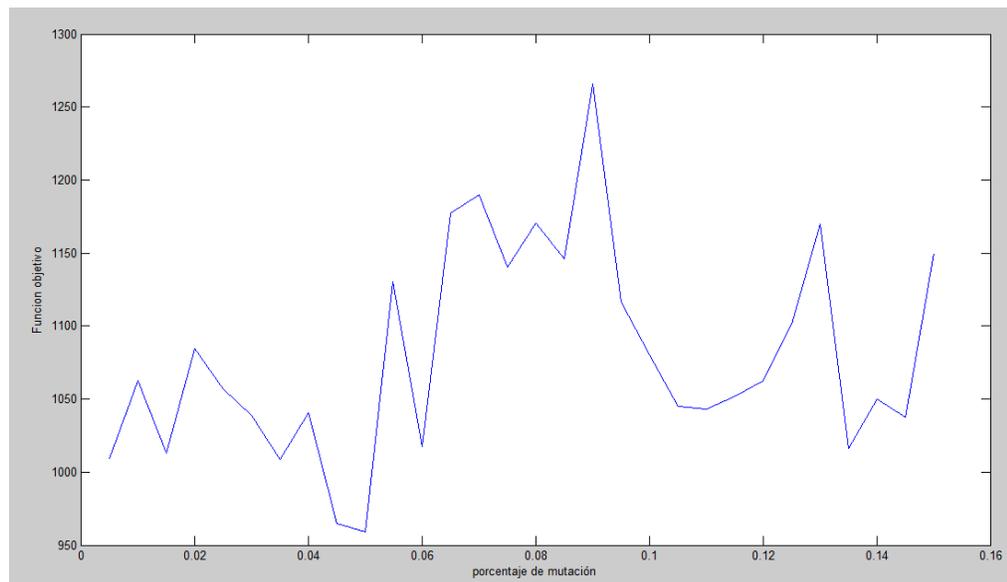


Figura 3-16. Comportamiento del porcentaje de mutación en el algoritmo genético.

3.1.6 Modificación de la población:

La población en cada generación se verifica el estado de los individuos, es decir que si estos presentan semejanzas entre sí, o que la función objetivo se mantiene constante, se debe introducir al menos un individuo con características no optimas que permita generar más individuos diferentes, con la posibilidad de empeorar la función objetivo, pero dando más espacio de soluciones a la población. Sin embargo se debe tener en cuenta que Si la alternativa no genera factibilidad se debe eliminar de la población, o eliminar al menos infectable de la población.

3.1.7 Etapa de mejoramiento de las soluciones:

Con el fin de mejorar las soluciones obtenidas en el algoritmo genético, se plantea tomar al mejor de la población y realizar la técnica de búsqueda de solución en entornos variables VNS. Finalmente se encuentran soluciones con menor error y se realiza una visualización de las rutas. Para mejorar aún más la solución se genera por medio del

algoritmo de intercambios 2OPT una solución al problema de TSP o agente viajero para cada ruta encontrada.

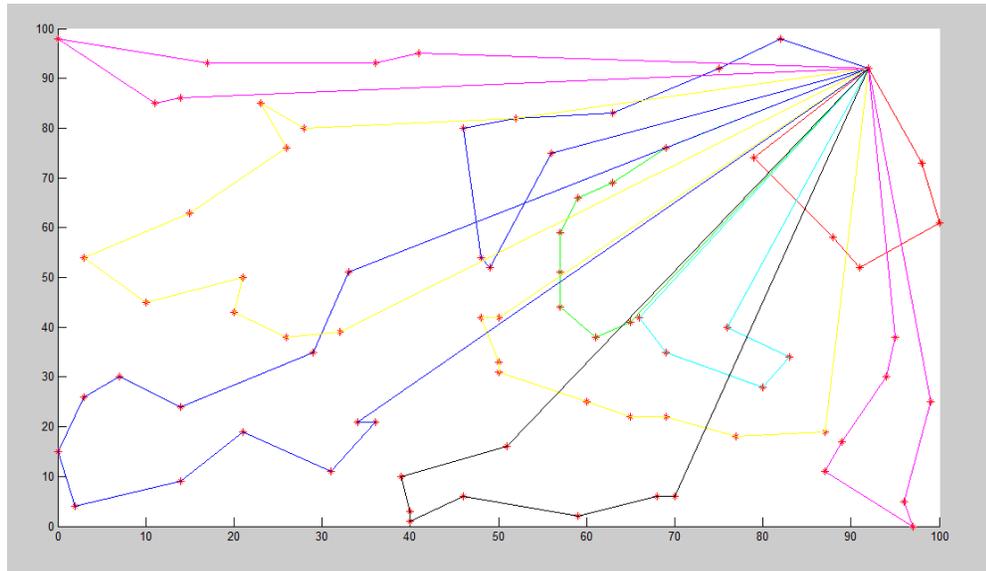


Figura 3-17. Rutas solución VRP instancia de Augerat para 80 ciudades.

La figura 3-17 indica las rutas solución del problema para una de las instancias publicadas como Augerat 80 en. [54].

3.2 IMPLEMENTACIÓN DE HEURÍSTICAS Y METAHEURÍSTICAS PARA LA CONFIGURACIÓN DE LA POBLACIÓN INICIAL DEL ALGORITMO GENÉTICO.

Las Heurísticas que se usan para generar la adición de individuos a la población inicial del algoritmo genético son las siguientes:

- Intercambio 2OPT
- Inserción del más barato
- Ruta mínima
- Técnica de ahorros
- Hormigas básicas
- Hormigas con intensificación
- Técnica del barrido

3.2.1 Heurística de intercambio 2opt.

El algoritmo consiste básicamente en eliminar dos aristas y conectar los dos caminos resultantes, de una manera diferente, para obtener un nuevo ciclo.

Un diagrama de flujo indica en términos generales el método de intercambio.

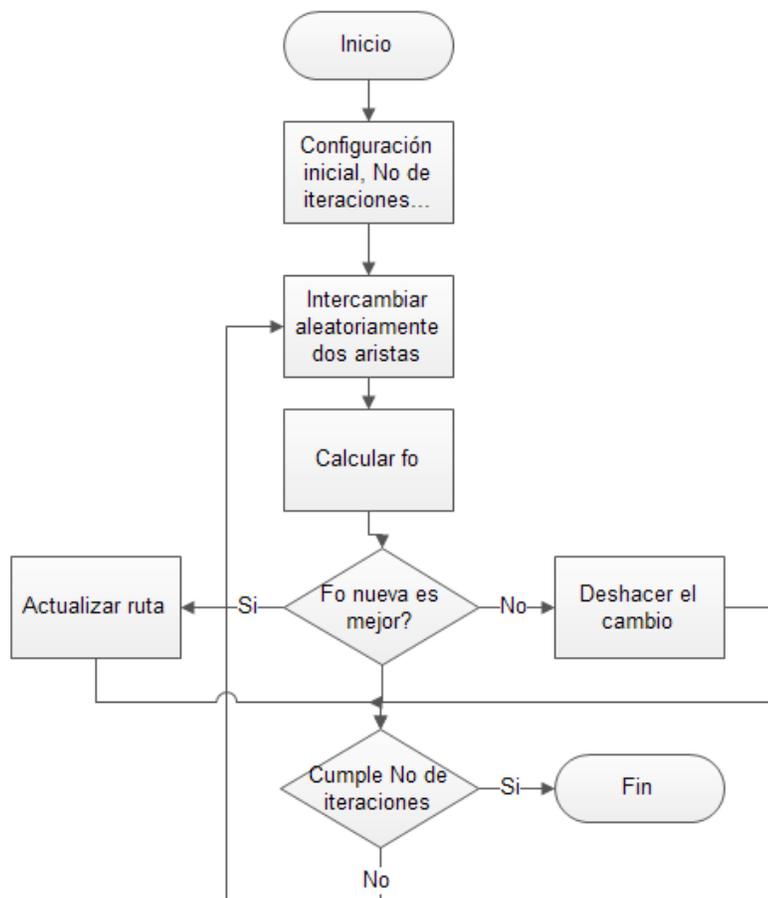


Figura 3-18. Diagrama de flujo para heurística 2 OPT

Para el caso particular del problema se toma un vector desde 2 hasta n , donde n representa el número de clientes que se debe visitar y el numeral 1 representa el depósito, por lo que no se indica en el vector, este vector se permuta para formar una ruta aleatoria y se genera para este las rutas respectivas de acuerdo con la capacidad de los vehículos y la demanda de los distintos clientes. De igual manera se obtiene el valor objetivo de cada ruta con el fin de verificar si los intercambios mejoran la calidad de las respuestas en función de la matriz de distancias.

Los intercambios se dan según los vectores de ruta y vehículos así por ejemplo en el caso de 10 clientes tenemos:

Vector permutado:

3	4	2	6	8	5	9	10	7
---	---	---	---	---	---	---	----	---

Figura 3-19. Vector permutado

Se determinan las rutas respectivas evaluando la capacidad del vehículo y generando el vector de vehículos y el del valor objetivo del vector permutado. El vector vehículos puede ser de la siguiente manera:

5	2	2	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---

Figura 3-20. Vector Vehículos

Este vector es de dimensión igual al número de clientes menos uno, es decir de la misma dimensión del vector de rutas, indicando que el primer vehículo visita las 5 primeras ciudades, en este caso las que corresponden a 3,4,2,6 y 8. El vehículo 2 visita las ciudades 5 y 9. El vehículo 3 visita las restantes, es decir la 10 y 7.

Con esto se pasa a realizar de forma aleatoria e iterada los respectivos cambios, los cuales se pueden realizar intercambiando dos nodos de cada ruta y verificando la función objetivo de cada una de las rutas o intercambiando dos nodos de toda la secuencia de visitas y con cada cambio asignar vehículos nuevamente para en un siguiente paso calcular el valor objetivo total y decidir si es adecuado dejar los cambios o continuar hasta terminar el número de iteraciones. Para este caso se toma la segunda opción con el fin de obtener respuestas rápidas y partiendo de que el vector inicial no es obtenido con algún algoritmo clásico de ruta mínima o ahorros, sino que es de forma aleatoria y la solución que se debe lograr es para alimentar una población inicial que en etapas posteriores se debe depurar.

3.2.2 Heurística de inserción del más barato.

Las heurísticas de inserción son métodos constructivos en los cuales se crea una solución mediante sucesivas inserciones de clientes en las rutas. En cada iteración se tiene una solución parcial cuyas rutas solo visitan un subconjunto de los clientes y se selecciona un cliente no visitado para insertar en dicha solución. Cuando la inserción se realiza de forma secuencial, se usa el algoritmo de Mole & Jameson en el que solo se considera insertar clientes en la última ruta creada, siendo esto una desventaja puesto que los últimos clientes no visitados tienden a estar dispersos y por lo tanto las últimas rutas construidas son de costo muy elevado. Un diagrama de flujo general sería como el que se indica a continuación:

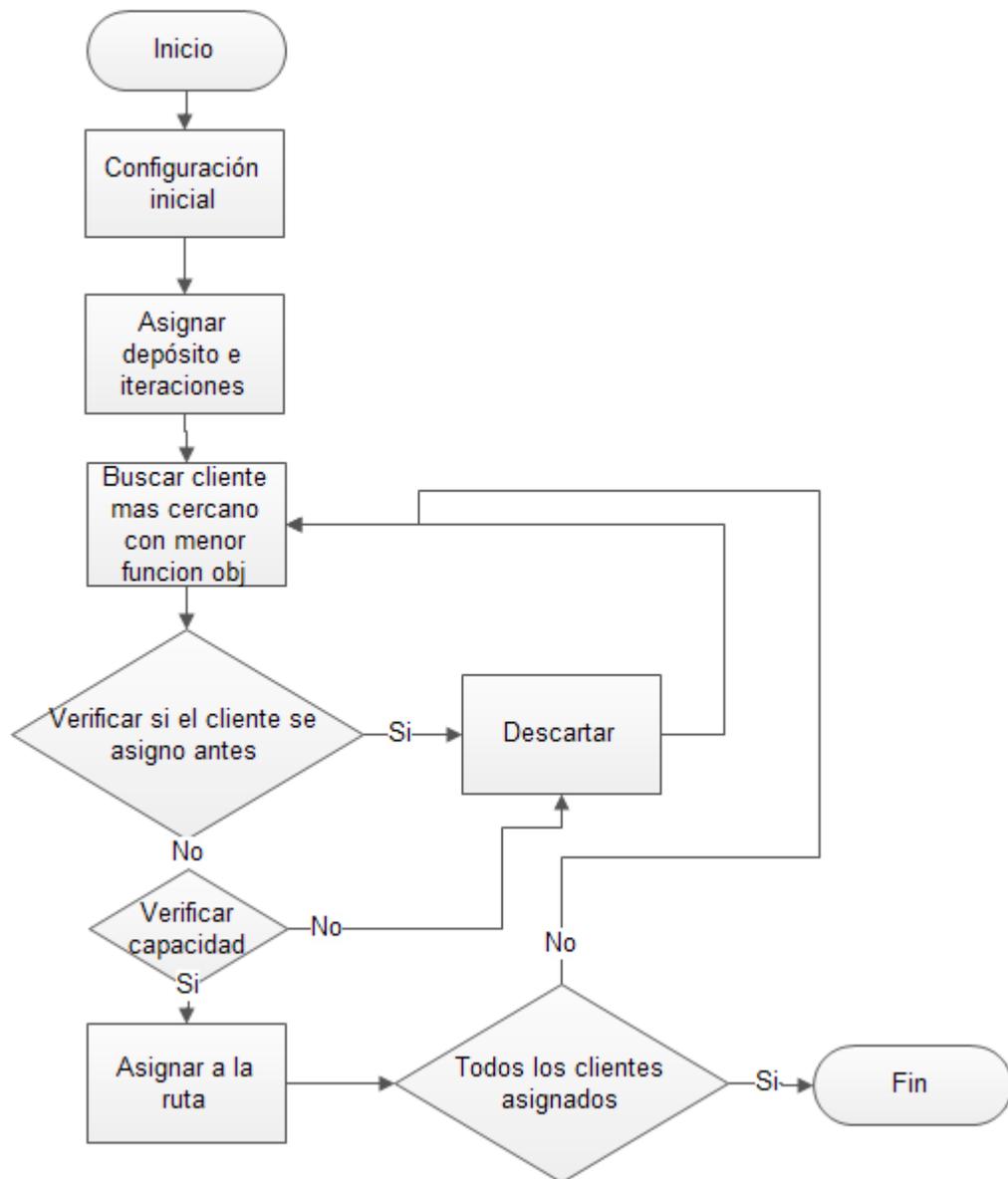


Figura 3-21. Diagrama de flujo para inserción del más barato

Otra forma de realizar las inserciones es en paralelo, de forma que se pueda insertar un cliente en cualquiera de las rutas de la solución para lo cual se puede usar el algoritmo de Christofides, Mingozzi y Toth, citados en la tesis de A. Olivera [30].

Para este caso por ser una solución de inicio se puede tomar el algoritmo de forma secuencial, a manera de ejemplo para 5 ciudades incluido el depósito se indica lo siguiente:

Q:

3	4	1	8
---	---	---	---

Figura 3-22. Vector demanda

C: 8.

D:

Tabla 3-6. Matriz de distancias para 5 ciudades

Ciudad	1	2	3	4	5
1	0	3	6	8	1
2	3	0	4	7	2
3	6	4	0	3	8
4	8	7	3	0	9
5	1	2	8	9	0

Donde Q representa el vector de demanda de las ciudades sin el depósito, puesto que este toma un valor de cero. C indica la capacidad de los vehículos y D es la matriz de distancias entre las ciudades.

Para llevar a cabo este algoritmo se genera una solución aleatoria como puede ser el siguiente vector de rutas:

Rutas:

2	5	4	3
---	---	---	---

Figura 3-23. Vector secuencia de clientes

Se forman las rutas para los vehículos dependiendo de la capacidad de estos y la demanda de la secuencia anterior.

Esto es que el vehículo 1, solo puede visitar a la ciudad 2, puesto que la 5 excede la capacidad del vehículo uno.

La ruta dos se forma con la visita a la ciudad 5 y termina, puesto que esta ciudad demanda la totalidad de la capacidad del vehículo dos.

La ruta tres se forma con la visita a las ciudades 4 y 3.

Como siguiente medida se pasa a revisar la posible inserción de una ciudad con el menor costo a la ruta uno, puesto que esta tiene disponible capacidad para visitar a otras.

Para no visitar ciudades que ya hayan sido insertadas se define un vector de dimensiones igual al de las demandas para marcar en la posición correspondiente con un valor de uno si ya fue insertada o con cero si aún no lo es.

Insertados:

0	0	0	0
---	---	---	---

Figura 3-24. Vector verificación de insertados

Al tomar la ruta dos se observa que se puede insertar a esta ruta las ciudades 5, 4 y 3, para lo cual se toma la de menor costo y verificando que cumpla con la restricción de capacidad del vehículo.

Así, la de menor costo es la ciudad 5 pero no cumple con la demanda del vehículo, por lo que se pasa a tomar otra ciudad, en este caso es la ciudad 3. Se actualiza el vector de insertados:

Insertados:

1	3	0	1
---	---	---	---

Figura 3-25. Vector verificación de insertados actualizado

Como aún existe capacidad en el vehículo se pasa a verificar si es posible insertar otra ciudad. La ciudad 4 tiene una demanda que es posible cubrir con esta ruta, luego se puede insertar, quedando la ruta uno así:

Ruta 1: 1-2-3-4-1

Con un costo de $3+4+3+8 = 18$

Como ya se cubrió todas las ciudades, se tiene el vector de rutas y vehículos para enviarlos a la población inicial del algoritmo genético así:

Rutas:

2	3	4	5
---	---	---	---

Figura 3-26. Vector de rutas

Vehículos:

3	1	0	0
---	---	---	---

Figura 3-27. Vector vehículos

El vector de vehículos indica que la ruta 1 debe visitar tres ciudades desde el depósito y volver (1-2-3-4-1). La ruta dos está dada por la visita a una ciudad (1-5-1).

El costo que se toma para la función objetivo es de $18+2=20$, que corresponde al costo de las dos rutas.

3.2.3 Heurística de la ruta mínima o vecino más cercano.

Esta heurística corresponde al método básico de solución al problema del agente viajero o TSP (Traveling salesman problem). De este modo se toma el número de clientes o ciudades del problema y se realiza una inspección inicial de la distancia mínima desde el depósito hacia los clientes, tomando como criterio visitar al que tenga la distancia mínima y así sucesivamente hasta completar la capacidad del vehículo para luego regresar al depósito y continuar con el siguiente vehículo hasta terminar de visitar a todos los clientes por medio del algoritmo de Dijkstra como se cita en la tesis de A. Olivera.[30].

En este caso se debe realizar un control sobre los clientes ya visitados y la capacidad de los vehículos, así se inicia con el nodo que tenga la distancia mínima con respecto al depósito y se continúa visitando el próximo más cercano al nodo anterior, cuando se haya terminado la capacidad del vehículo se vuelve a calcular la distancia mínima de los clientes no visitados con respecto al depósito y se continúa el proceso colocando un valor de uno en un vector de dimensión igual al número de ciudades para indicar que en la posición de dicha ciudad ya se hizo la visita, con lo que se restringe a ser visitado nuevamente por otro vehículo. El siguiente diagrama de flujo general muestra en forma resumida el proceso.

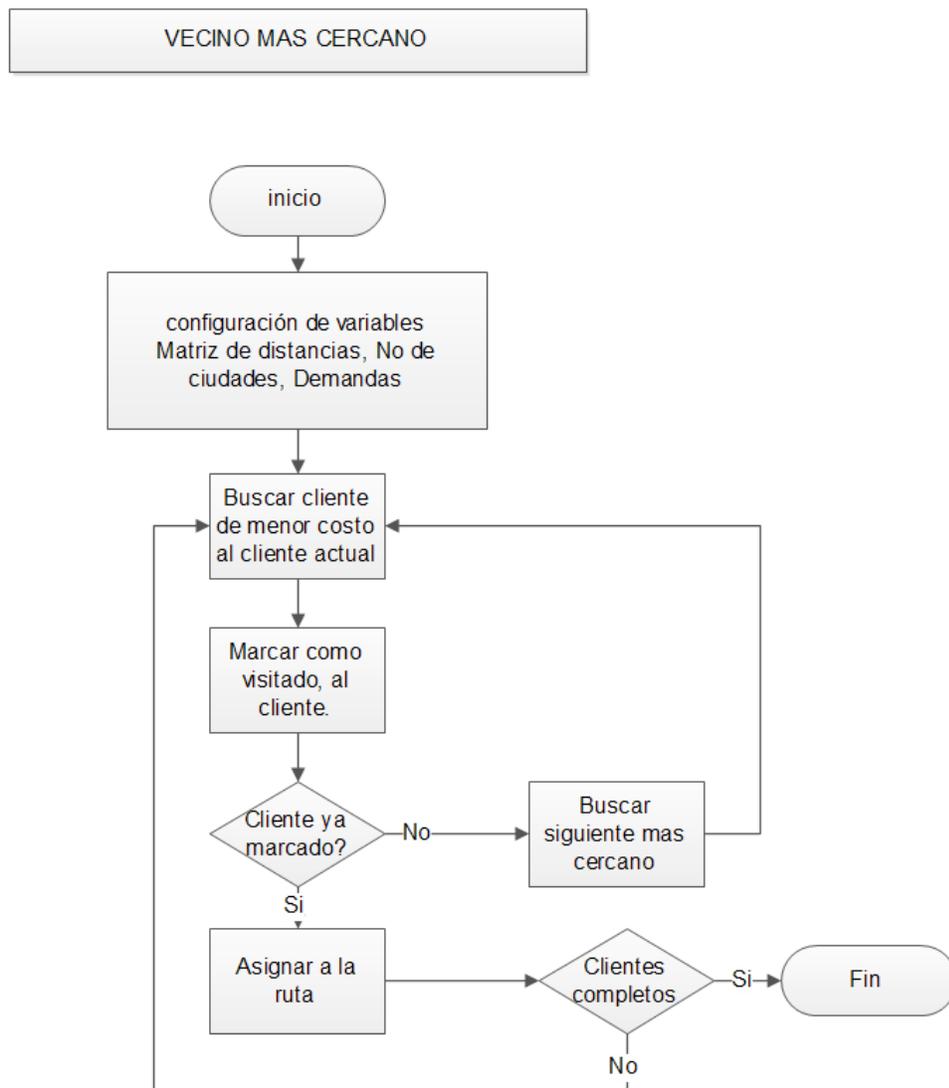


Figura 3-28. Diagrama de flujo del vecino más cercano

Por ejemplo si se tiene 5 ciudades, con un vector de demanda Q de dimensión igual al número de ciudades menos uno ya que el depósito no tiene demanda. Vehículos con capacidad C , y una matriz de distancias D , con dimensión 5×5 , donde la fila uno y columna uno indica la posición del depósito.

A manera de ejemplo se puede tomar los vectores anteriormente mencionados de la siguiente manera:

Q:

3	4	1	8
---	---	---	---

Figura 3-29. Vector demanda

C: 8.

D:

Tabla 3-7. Matriz de distancias para 5 ciudades

Ciudad	1	2	3	4	5
1	0	3	6	8	1
2	3	0	4	7	2
3	6	4	0	3	8
4	8	7	3	0	9
5	1	2	8	9	0

Se puede observar que el punto de partida para la ruta es el de menor distancia al depósito, en este caso es la ciudad 5 con distancia de 1 y demanda de 8.

El siguiente paso es verificar si el vehículo puede satisfacer a este cliente o no, si es así se adiciona a la ruta.

En este caso como la demanda de esta ciudad es igual a la capacidad del camión, la ruta uno queda desde el depósito hasta la ciudad 5 con regreso al depósito. 1-5-1 con demanda igual a 8 y se actualiza el vector de ciudades visitadas V.

V:

0	0	0	1
---	---	---	---

Figura 3-30. Vector verificación visitas

El vector anterior es de dimensión igual al número de ciudades menos uno, puesto que el depósito no se cuenta.

Luego se identifica el cliente o ciudad con menor costo desde el anterior visitado y teniendo en cuenta que no se puede considerar la distancia de las ciudades que tengan en el vector V el valor de uno. es decir desde la ciudad 5. En este caso es la ciudad 2. Ahora se verifica la capacidad del vehículo que en este caso no es posible visitar a otra ciudad, por lo cual se toma un nuevo vehículo con la distancia mínima desde el depósito sin incluir la ciudad visitada 5.

Como se observa en la matriz de distancias el valor mínimo es el que tiene la ciudad 2 con un valor de 3.

Se continúa con el mismo método, en el cual se obtiene la distancia mínima desde la ciudad 2 a la ciudad 3 con un valor o costo de 4 unidades, se verifica que el vehículo tenga capacidad y se continúa desde la ciudad 3 a la ciudad 4 con costo de 3 unidades. Cave notar que con cada ciudad visitada, si es posible según la capacidad del vehículo se debe marcar si es o no visitada en el vector de visitas V .

Según el orden de visitas se tiene V :

1	0	0	1
1	1	0	1
1	1	1	1

Figura 3-31. Orden de visitas

Al tiempo se puede ir sumando el costo de la ruta. Así la ruta uno es:

1 – 5 – 1, con un costo de $1+1=2$.

La ruta dos:

1 – 2 – 3 – 4 – 1, con costo de: $3+4+3+8= 18$.

El costo total es el costo del total de rutas: $2+18=20$.

La solución que se envía como individuo de la población del algoritmo genético en este caso es el vector de rutas, el valor objetivo y el vector de vehículos con la dimensión igual al número de ciudades menos uno así:

Ruta:

5	2	3	4
---	---	---	---

Figura 3-32. Vector rutas

Vehículos:

1	3	0	0
---	---	---	---

Figura 3-33. Vector vehículos

Función objetivo: 20.

3.2.4 Heurística de ahorros de Clarke y Wriht

La idea central de las Heurísticas de Ahorro es que se van combinando las rutas, en la medida que al pasar a ser una sola ruta se producen ahorros de costos. Uno de los algoritmos de este tipo, y a la vez de los más difundidos para el VRP es el Algoritmo de Ahorros de Clarke & Wright. Citados en la tesis de A. Olivera . [30]. Si en una solución dos

rutas diferentes $(0, \dots, i, 0)$ y $(0, j, \dots, 0)$ pueden ser combinadas formando una nueva ruta $(0, \dots, i, j, \dots, 0)$.



Figura 3-34. Grafo para Técnica de ahorros.

El ahorro en distancia obtenido por dicha unión se presenta con la siguiente formula:

$$s_{ij} = c_{i0} + c_{0j} - c_{ij} \quad (3-1)$$

Donde s representa el ahorro de introducir la ruta i,j . La variable c , representa el costo de ir desde i al origen, o de j al origen y de i a j .

La nueva solución obtenida es la que genera el mayor ahorro siempre que no se violen las restricciones del problema como puede ser la capacidad del vehículo, o el sentido de las vías, entre otras.

Los pasos a seguir en esta metodología son los siguientes:

1. Para cada cliente i construir la ruta $(0, i, 0)$.
2. Calcular una matriz de ahorros.
3. Tomar el máximo ahorro y verificar si la solución es factible. Si lo es tomar como nueva solución, de lo contrario volver a calcular ahorros con otro cliente.

Una de las deficiencias del algoritmo de Clark & Wright es que tiende a producir buenas rutas al comienzo, pero rutas menos interesantes hacia el final. Para remediar esto, se puede realizar una penalización en la función de ahorros con la unión de rutas con clientes lejanos. Notaciones que se encuentran en la tesis de A. Olivera, [30].

El siguiente diagrama de flujo indica el proceso de manera resumida:

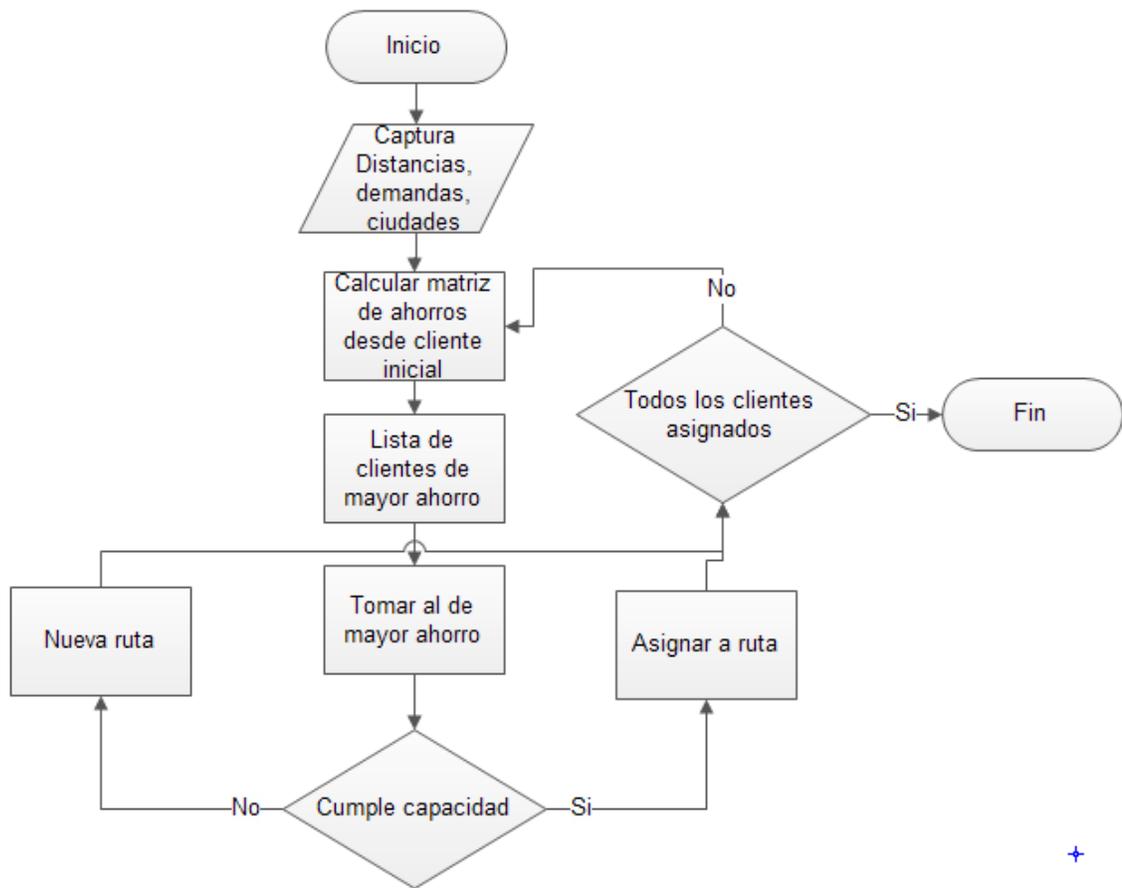


Figura 3-35. Diagrama de flujo técnica de ahorros

Como ejemplo se indica el problema de 5 ciudades con costos de distancias en una matriz D y vehículos con capacidad igual a 8 unidades y un vector de demanda Q así:

Q:

3	4	1	8
---	---	---	---

Figura 3-36. Vector demanda

C: 8.

D:

Tabla 3-8. Matriz de distancias para 5 ciudades en técnica de ahorros

Ciudad	1	2	3	4	5
1	0	3	6	8	1
2	3	0	4	7	2

3	6	4	0	3	8
4	8	7	3	0	9
5	1	2	8	9	0

Como primera medida se tendría una ruta para cada ciudad, es decir que existen 4 rutas como se indica en la figura 3-8:

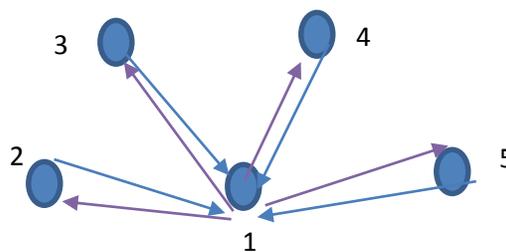


Figura 3-37. . Asignación de ruta para cada nodo

Con esto, se pasa a encontrar la matriz de ahorros usando la *fórmula (3.1)*:

$$s_{ij} = c_{i0} + c_{0j} - c_{ij}$$

$$s_{23} = c_{21} + c_{13} - c_{23} = 3 + 6 - 4 = 5$$

$$s_{24} = c_{21} + c_{14} - c_{24} = 3 + 8 - 7 = 4$$

$$s_{25} = c_{21} + c_{15} - c_{25} = 3 + 1 - 2 = 2$$

$$s_{34} = c_{31} + c_{14} - c_{34} = 6 + 8 - 3 = 11$$

$$s_{35} = c_{31} + c_{15} - c_{35} = 6 + 1 - 8 = -1$$

$$s_{45} = c_{41} + c_{15} - c_{45} = 8 + 1 - 9 = 0$$

Según lo calculado, se toma las rutas de mayor ahorro, en este caso es la que corresponde a la ruta de 3 a 4 con un ahorro de 11 unidades.

Ahora se verifica si el vehículo puede satisfacer la demanda, en este caso la ruta 1-3-4-1 tiene una demanda de $4+1=5$ unidades, lo que indica que el vehículo si puede satisfacer esa demanda, por lo que la ruta es válida, quedando de la siguiente forma:

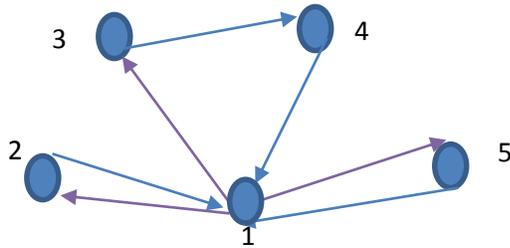


Figura 3-38. Introducción de nodo de máximo ahorro a la ruta

Nuevamente se calcula los ahorros en las rutas anteriores:

$$s_{34,5} = c_{34,1} + c_{15} - c_{34,5} = 6 + 3 + 8 + 1 - 9 = 9$$

$$s_{34,2} = c_{34,1} + c_{12} - c_{34,2} = 6 + 3 + 8 + 3 - 8 = 12$$

En este caso el máximo ahorro se presenta al adicionar el cliente 2 a la ruta 1-3-4-2-1. Se verifica la capacidad del vehículo que es de $4+1+3=8$, que en este caso si es posible, de lo contrario se tendría que verificar con otra ruta.

Finalmente las rutas quedan así:

Ruta1: 1-3-4-2-1

Ruta 2: 1-5-1

El valor de la función objetivo para este caso es de:

Ruta 1: $6+3+7+3=19$

Ruta 2: $1+1=2$

El valor objetivo que se pasa a la población de los algoritmos genéticos es de 21.

El vector de rutas es:

3	4	2	5
---	---	---	---

Figura 3-39. Vector ruta

El vector de vehículos es:

3	1	0	0
---	---	---	---

Figura 3-40. Vector vehículos

3.2.5 Técnica del barrido.

Esta técnica hace una asignación inicial para luego generar la ruta. Básicamente consiste en tomar las coordenadas de la posición angular de los clientes y hacer un barrido angular desde el ángulo más cercano al depósito, se hace girar una recta con origen en el depósito hacia los clientes, estos se van asignando a la ruta mientras se cumpla la restricción de capacidad del vehículo, además si existen dos o más clientes con el mismo ángulo de orientación, se ordena de acuerdo al más cercano según el radio. Se puede hacer una mejora de las rutas usando algoritmos de vecino más cercano u otros para resolver el TSP en cada ruta.

El diagrama de flujo indica en resumen esta técnica:

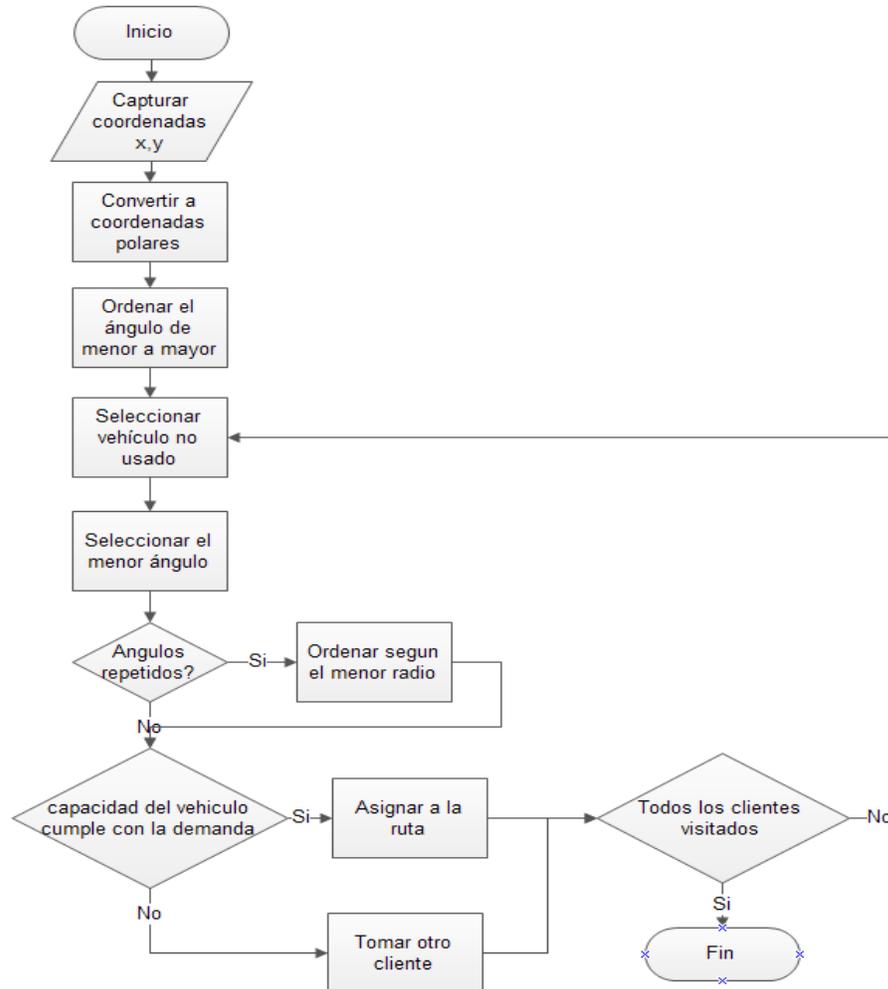


Figura 3-41. Diagrama de flujo para técnica del barrido

En el caso particular se usa esta técnica para encontrar una solución que se introduce a la población inicial del algoritmo genético.

3.2.6 Metaheurística de Colonia de hormigas.

Según el capítulo 8 del libro de técnicas metaheurísticas publicado en el año 2008 por A. Escobar, et al [38]. La metaheurística Optimización mediante Colonias de Hormigas o Ant Colony Optimization propuesta para resolver problemas complejos de optimización combinatoria, tiene su fuente de inspiración en el comportamiento de colonias de hormigas reales y fue formulada por primera vez en el año de 1996 en un problema de optimización combinatorial.[38].

El algoritmo para alimentar la población inicial de los genéticos consiste básicamente en tomar un vector aleatorio para el agente viajero con el número total de ciudades del problema de ruteo de vehículos y teniendo en cuenta que el deposito corresponde a la ciudad número 1, se procede a generar las soluciones para 5 hormigas con probabilidad de evaporación de feromona del 50% y con 1000 iteraciones, (parámetros que permiten obtener soluciones con bajo costo computacional para alimentar población inicial de genéticos). Se calcula el valor de la función objetivo y se compara con el de un valor incumbente inicializado en un número muy alto que se va actualizando con las iteraciones en la medida en que las hormigas confluyan a mejores soluciones. Al finalizar el algoritmo de las hormigas se toman las soluciones de 5 hormigas en el TSP y se procede a calcular las rutas con el criterio de capacidad de los vehículos. Finalmente se pasa los vectores de rutas y vehículos junto con el valor objetivo de las 5 hormigas a la población inicial del genético.

3.2.7 Metaheurística Colonia de hormigas con intensificación

En este caso el algoritmo tiene la misma configuración del caso anterior, usando la metodología de la colonia de hormigas, de igual manera se toma una solución aleatoria para el problema del agente viajero y a esta solución se generan las rutas con el criterio de la capacidad del vehículo, la diferencia está en que las soluciones se seleccionan por el método de la ruleta y no de forma aleatoria desordenada, a esto también se adiciona que el cambio de las ciudades para la generación de nuevas soluciones se hace por medio del algoritmo de intercambio 2OPT. Posterior a esto se toma la solución al problema del agente viajero y se generan las rutas para ingresar a la población inicial del algoritmo genético el vector de rutas de las 5 hormigas, sus vehículos y el valor objetivo de dichas rutas.

3.2.8 Intensificación algoritmo genético usando Metaheurística VNS.

Luego de generar la segunda población usando los diferentes algoritmos mencionados anteriormente, se procede a realizar las etapas del algoritmo genético como son la selección, cruce y mutación, creando varias generaciones con la población mejorada, esto da como resultado un mejoramiento en las soluciones, que finalmente se toma la que tenga el menor valor en la función objetivo para realizar una búsqueda por medio del algoritmo de Búsqueda en vecindario o entorno variable denominada VNS (variable neighborhood search) que consiste en hacer una búsqueda de mejoras sobre los óptimos locales.

Para el caso de mejoramiento de la solución alcanzada por el algoritmo genético se realiza el movimiento intercambiando vecinos del vector de rutas y verificando si los cambios mejoran la solución y teniendo en cuenta la capacidad de los vehículos. Estos movimientos se muestran en un ejemplo con 5 ciudades como se indica a continuación:

Ruta:

1	2	3	4	5
---	---	---	---	---

Figura 3-42. Vector ruta

Vecinos:

2	1	3	4	5
3	2	1	4	5
4	2	3	1	5
5	2	3	4	1

1	3	2	4	5
1	4	3	2	5
1	5	3	4	2

1	2	4	3	5
1	2	5	4	3

1	2	3	5	4
---	---	---	---	---

Figura 3-43. Combinaciones VNS

De esta manera se hace una búsqueda de la solución mejorando de manera general el vector de todas las ciudades del algoritmo.

Finalmente al tener la solución más aproximada al óptimo global se procede a realizar un intercambio 2OPT en las rutas obtenidas en la solución del VNS, tomando el problema como varios agentes viajeros, para lograr obtener el óptimo de cada ruta.

4. CARACTERIZACION DE RUTAS, MATRIZ DE COSTOS Y DEMANDAS PARA RUTAS DE LA EMPRESA COLÁCTEOS EN LA CIUDAD DE SAN JUAN DE PASTO (NARIÑO).

En este capítulo se describe inicialmente a la empresa COLACTEOS, su contexto y los procesos para determinar las condiciones que presenta la empresa para la distribución de sus productos en la ciudad de San Juan de Pasto, contando con un número definido de vehículos y la cantidad de tiendas clientes.

Por otra parte se indica el proceso de obtención de la matriz de distancias entre los puntos geo referenciados a los que atiende la empresa, el valor de la demanda de cada punto de acuerdo con el valor de despachos que realiza, y el valor de costos de rutas que se tienen en el momento. Para esto se procede a realizar una consulta del estado actual de la distribución, posteriormente se realiza una búsqueda de sectores a los que atiende cada vehículo y se obtiene las coordenadas de cada punto para en seguida calcular la distancia entre ellos, su demanda en unidades de carga de cada vehículo hacia cada sector de las rutas.

4.1 DESCRIPCIÓN DE LA EMPRESA COLACTEOS

La empresa COLÁCTEOS, [55] hacia la década de los años 70, los ganaderos de Nariño, deciden unirse y sacar adelante la ganadería, organizándose para poder exportar leche a mejores precios y hacia 1974 nace la idea de fundar una Cooperativa.

Fue entonces como el 2 de Marzo de 1977, cuando 30 ganaderos de la región quienes ya habían madurado la idea de agremiarse y con el apoyo y colaboración de la misión técnica del Gobierno Holandés, se firmó el Acta de constitución de la Cooperativa de Productos Lácteos de Nariño, con domicilio en la ciudad de Pasto.

Han transcurrido ya más de 30 años desde su fundación, cumpliendo con uno de los principales fines para el cual fue creada:

Agrupar a ganaderos del departamento, garantizándoles perspectivas de desarrollo en su campo de acción. Hoy en día los objetivos propios de la Cooperativa contemplados en sus Estatutos se definen al tenor de los principios cooperativos, el bienestar económico y social del gremio ganadero de Nariño.

La Cooperativa se inicia con la Planta Enfriadora de Guachucal donde se capta leche e inicialmente se la enfría para venderla a otros departamentos.

Hacia 1979 se hace el primer intento de pasterizar leche y se lanza al mercado la leche pasteurizada Ñapanguita, pero la falta de experiencia en el manejo y mercadeo hacen que este intento no fructifique, en este momento es fundamental el apoyo de la Misión Holandesa con su técnica y conocimientos.

En julio de 1984, se lanza nuevamente al mercado leche pasteurizada de Coprolácteos, inicialmente en Pasto y dada su aceptación y demanda se extendió a todo el departamento de Nariño y en la actualidad hacia el departamento del Cauca y Valle.

En el cuatrienio 84-88 la Cooperativa amplía su planta de procesamiento de quesos y derivados y en el primer trimestre del año 87, inicia la producción de yogur y kumis en presentaciones de vaso y garrafa.

Hacia el año 1985-1986, la Cooperativa inicia en Cali el expendio de leche Colácteos en dicha ciudad, al adquirir una planta pasteurizadora.

A partir del año 1993, se inicia un proceso administrativo que se enmarca en seis aspectos fundamentales:

- Consolidación de la Estructura Organizacional Administrativa y Financiera de la Cooperativa.
- Consolidación del Mercado con proyección nacional e internacional.
- Construcción y remodelación de la Estructura Básica Industrial.
- Comercialización con competitividad y eficiencia.
- Planeación y Desarrollo de nuevos Productos.
- Proyectos de Mejoramiento y proyección institucional.

Es así como se logra el desarrollo e introducción en nuevos mercados de los productos: Quesillo, Queso Mozzarella, Yogur con Cereal, Queso Campesino Semiduro, Queso Fresco, Queso Fundido, Yogur en Bolsa, Yogur con Café, Quesito.

Hacia el año de 1997 se adquiere la marca Tapioka, procesando toda la línea de fermentados con esta marca, así como el Queso Cuajada Tapioka, Quesillo semigraso, Queso Tipo Sándwich, Yogur sin Azúcar, Jugos y Agua. En el año 1997 se realiza la inauguración de una nueva Planta ubicada en el municipio de Pupiales, destinada a la producción de quesos frescos e hilados.

En esta década toma auge el desarrollo de las marcas blancas, es decir las MAQUILAS, Colácteos trabaja esta figura para clientes tan importantes como Rica Rondo, Cafam, Olímpica, Aro (Makro), Cadenalco -Éxito.

El portafolio de productos, con que cuenta actualmente la Cooperativa lo constituye:

- Leche Pasteurizada
- Yogur
- Yogur Cereal
- Kumis
- Crema de Leche
- Arequipe
- Quesos Maduros, Parmesano, gratinar, holandés, gouda, sandwichero.

- Quesos Frescos: Campesino, Quesillo, Doble Crema, fundido industrial, fresco, molido.
- Mantequilla
- Jugos
- Agua

Estos productos se fabrican en las diferentes plantas con las que Colácteos cuenta ubicadas en Guachucal, Pasto, Pupiales, y Cali.

El fin de este proyecto se enmarca en la adecuación de una metodología adecuada para la distribución de los distintos productos que tienen la planta de Colácteos en el ciudad de Pasto, cuya metodología actualmente se desarrolla con base en el empirismo de los conductores de los 25 vehículos contratados por la empresa, los cuales ingresan a la planta todos los días para cargar el producto desde las 4:00 am y se hace por orden de llegada del conductor. La empresa cuenta con dos carriles en los cuales los vehículos se estacionan para cargar el producto que el día anterior fue solicitado por el conductor según la demanda de los clientes que en general se mantiene constante según la información que se obtiene de los datos estadísticos de la empresa. Los productos solicitados se arruman y se sacan hasta el carril de carga de la planta y en la medida en que los vehículos finalizan la carga, estos salen a distribuir los productos a las tiendas que cada conductor tiene asignado como clientes y que visita únicamente por experiencia y no por rutinas de recorridos obtenidas con metodologías adecuadas u obtenidas por algún tipo de modelación matemática.

4.2 RUTAS INICIALES POR VEHICULOS DE LA EMPRESA COLÁCTEOS EN LA CIUDAD DE SAN JUAN DE PASTO.

Según entrevista a los conductores de los vehículos que cubren la distribución de los productos lácteos en la ciudad de San Juan de Pasto, se plantean las rutas por sectores para cada vehículo así:

Empresa: Existen dos tipos de distribuidores, uno que pertenece a la empresa Colácteos, la cual distribuye todos los productos de la marca Colácteos y otro que pertenece a la empresa Tapioca con productos que empaca Colácteos pero con la marca Tapioca.

La carga de los vehículos está dada en número de cubetas, las cuales contienen bolsas de leche de litro y 750 cc. Otro tipo de carga es el número de canastas, las cuales contienen otros productos como queso, arequipe, yogurt.

El proceso de carga se hace en el parqueadero de la empresa, en donde el día anterior cada propietario del vehículo realiza el pedido y la empresa alista y arruma estos pedidos para al día siguiente realizar el despacho desde las 4:00 am, siguiendo la rutina de carga por turnos de 15 minutos en 4 carriles como lo indica el registro fotográfico siguiente:



Figura 4-1 Forma de carga de productos en vehículos de la empresa Colácteos.

Según la información recolectada en la empresa se tiene un resumen de los datos de cada vehículo así:

Vehículo 1.

Propietario: Hernando Rodríguez.

Empresa: Colácteos

Hora de llegada: 4:00 am

Hora de salida: 4:20 am

Tiempo de recorrido: 9 horas.

Número de tiendas: 120.

Costo por combustible diario: \$ 15 000.

Carga cubetas: 60.

Carga Canastos: 30

Ruta:

Tabla 4-1. Ruta del vehículo 1 de Colácteos por sectores y coordenadas de latitud y longitud

Indicador mapa	Latitud, Longitud	Lugar Ruta
A	1.24788,-77.265499	Planta Colácteos Aranda
B	1.202142,-77.284454	Caicedo
C	1.200598,-77.282517	Granada
D	1.200018,-77.281118	ciudad jardín
E	1.199182,-77.282492	Nueva Colombia
F	1.197626,-77.277115	María Isabel (Estadio libertad)
G	1.200362,-77.283359	Chapál.
H	1.20107,-77.270046	Las Violetas
I	1.205596,-77.277342	Aire libre
J	1.207806,-77.279598	Av Las Américas
K	1.204309,-77.280475	Atahualpa
L	1.204845,-77.280453	San Miguel
M	1.24788,-77.265499	Colácteos.

La ruta que indica el conductor del vehículo se genera en el mapa de la ciudad de San Juan de Pasto haciendo uso de la aplicación de googlemaps como se indica en la figura 4-2.

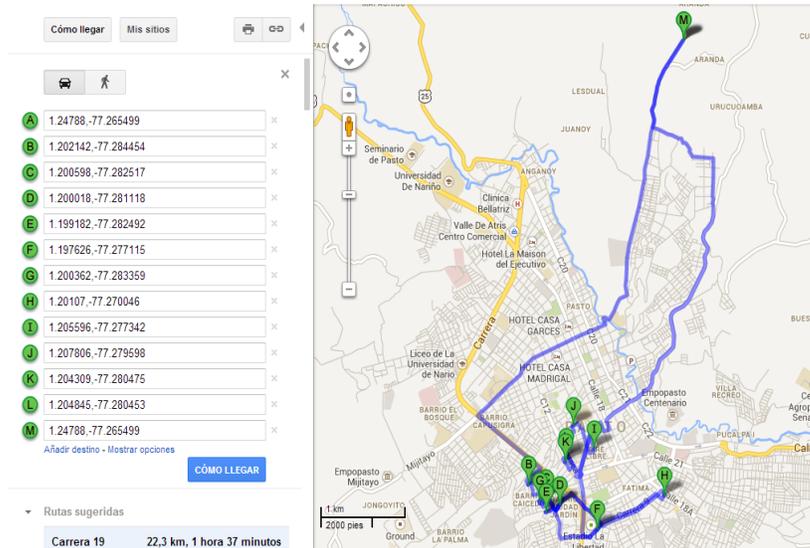


Figura 4-2. Ruta del vehículo 1 obtenido en la aplicación de googlemaps

En el recorrido indicado en la figura 4-2 se obtiene las coordenadas de los sectores visitados en latitud y longitud, junto con el valor de distancia y tiempo que se tardaría en el recorrido de forma promedio sin tener en cuenta congestiones ni paradas por el vehículo. En este caso se nota que existe una distancia de 22,3 Km para un tiempo de 1 hora con 37 minutos de lo que se puede obtener un valor de velocidad media para rutas sin congestión vehicular de 13,79Km/h.

De igual forma se obtiene las coordenadas, el tiempo y la distancia de las rutas de los demás carros.

Como un indicativo de la velocidad a la cual se mueven los vehículos según la aplicación de googlemaps, se muestra el promedio de la velocidad, el cual para este caso es de **13.88 Km/h.**

Al igual el valor promedio de tiendas a visitar cada ruta es de aproximadamente **106.**

El tiempo que gasta cada vehículo en cubrir la ruta realmente está alrededor de 8 horas y el gasto en combustible es cerca de \$16000.

De igual forma se toma los sectores con los valores de coordenadas en latitud y longitud correspondientes para cada punto. Cabe resaltar que en algunos casos los vehículos pueden ir a puntos iguales, por lo que se debe tener en cuenta tomar únicamente los sectores sin repeticiones, para lo cual se encuentra que las rutas tienen 154 sectores incluyendo el depósito.

4.3 ASIGNACION DE DEMANDA A CADA SECTOR.

El valor de la demanda se realiza con base en los sectores que visita cada ruta, para ello se debe codificar los 154 sectores e identificar a cuales de estos le aporta cada una de las 24 rutas que se tienen como información de la empresa COLÁCTEOS, para esto se usa una búsqueda de las coordenadas de cada ruta en la matriz que contiene los 154 puntos y se codifica con un número que indique a que sector se hace el aporte de demanda en cada ruta.

Tabla 4-2. Resultados de los sectores correspondientes a cada vehículo

r1	r2	r3	r4	r5	r6	r7	r8	r9	r10	r11	r12	r13	r14	r15	r16	r17	r18	r19	r20	r21	r22	r23	r24
código de sector																							
2	13	24	32	41	51	62	74	27	32	84	91	27	107	115	120	126	24	25	123	110	16	82	26
3	14	25	33	42	26	63	71	60	39	85	92	96	49	108	121	127	25	134	138	107	147	30	151
4	15	26	34	43	27	64	75	26	32	51	93	80	108	109	122	128	26	135	139	49	148	28	152
5	16	27	35	44	52	65	73	52	50	86	94	104	109	2	103	85	27	136	140	145	13	149	88
6	17	28	36	45	53	66	67	53	80	87	95	38	110	116	91	129	133	137	141	146	21	150	153
7	18	29	37	46	54	67	108	54	81	88	96	39	30	117	101	88	30	0	142	145	22	0	111
8	19	30	38	47	55	68	50	55	82	89	97	105	111	118	123	130	25	0	24	109	0	0	54
9	20	31	39	48	56	69	14	57	83	90	98	9	112	119	124	131	0	0	143	0	0	0	130
10	21	25	40	49	57	70	23	0	9	0	99	106	113	7	125	132	0	0	144	0	0	0	154
11	22	0	0	50	58	71	64	0	25	0	100	0	114	0	0	0	0	0	145	0	0	0	85
12	23	0	0	67	59	72	76	0	0	0	101	0	52	0	0	0	0	0	0	0	0	0	0
0	24	0	0	0	60	73	77	0	0	0	102	0	27	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	61	50	78	0	0	0	103	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	62	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	79	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

En la tabla 4-2, se tiene el valor del sector que atiende cada uno de los vehiculos, asi por ejemplo el vehiculo 19, atiende sin incluir el deposito a 5 sectores codificados con los numeros 25, 134, 135, 136 y 137, los cuales corresponden a los sectores etiquetados como Las cuadras, Parque infantil, Centro (Éxito del centro), Misterpollo (Centro) y Pollo sorpresa (centro).

Con los sectores que atiende cada ruta se genera el valor de demanda por sector, tomando unidades de cantidad de canastos por tiendas y por sectores, teniendo en cuenta que el depósito no se considera como sector ya que no requiere demanda.

Tabla 4-3. Demanda de canastos por tiendas y por sectores de cada vehículo.

Vehículo	No de sectores	No de canastos	No de tiendas	canastos/No tiendas	No de canastos/sector	No tiendas/sector	(canastos/tienda)/sector
1	12	60	120	0,5	5,45	10,91	0,04545

2	12	52	120	0,43	4,73	10,91	0,03939
3	10	35	70	0,5	3,89	7,78	0,05556
4	10	70	220	0,32	7,78	24,44	0,03535
5	12	40	50	0,8	3,64	4,55	0,07273
6	14	86	120	0,72	6,62	9,23	0,05513
7	15	82	200	0,41	5,86	14,29	0,02929
8	16	40	180	0,22	2,67	12	0,01481
9	9	43	80	0,54	5,38	10	0,06719
10	11	25	70	0,36	2,5	7	0,03571
11	9	71	135	0,53	8,88	16,88	0,06574
12	14	50	120	0,42	3,85	9,23	0,03205
13	10	28	78	0,36	3,11	8,67	0,03989
14	13	50	86	0,58	4,17	7,17	0,04845
15	10	58	150	0,39	6,44	16,67	0,04296
16	10	38	60	0,63	4,22	6,67	0,07037
17	10	38	60	0,63	4,22	6,67	0,07037
18	8	35	60	0,58	5	8,57	0,08333
19	6	28	70	0,4	5,6	14	0,08
20	11	45	50	0,9	4,5	5	0,09
21	8	53	100	0,53	7,57	14,29	0,07571
22	7	47	150	0,31	7,83	25	0,05222
23	6	40	80	0,5	8	16	0,1
24	11	80	110	0,73	8	11	0,07273

Para encontrar el valor de canastos que demanda cada sector y teniendo en cuenta que un sector puede ser atendido por varios vehículos, se considera la demanda de cada sector así:

$$\text{Demanda del sector } x \text{ en canastos} = (\text{Valor de canastos/tienda/sector}) * (\text{suma de tiendas de los vehículos que atienden al sector } x) \quad (4-1)$$

Para encontrar el valor de la demanda de cada uno de los 154 sectores se desarrolla una búsqueda inicialmente de los vehículos que le aportan con su visita a cada sector y posteriormente se obtiene el valor de la demanda de cada sector con el aporte que realice cada ruta a cada sector.

4.4 CALCULO DE MATRIZ DE DISTANCIAS ENTRE SECTORES PARA CARACTERIZAR COSTOS.

Con la información obtenida de las rutas que hace la empresa para sus vehículos, se procede a ingresar la información a la web con el fin de obtener la distancia entre cada

uno de los puntos referidos como barrios o sectores a los cuales atiende la empresa de distribución de productos lácteos, para esto se hace uso de la programación en un entorno JAVA y HTML solicitando el servicio de cálculo de distancia entre puntos en la API de googlemaps. Cabe resaltar que la distancia que calcula el googlemaps es el valor real teniendo en cuenta el sentido de las calles, cruces permitidos y posición geográfica para pendientes rectas; esto gracias a la asignación de las direcciones por medio de coordenadas escritas en latitud y longitud, las cuales han sido geo referenciadas de acuerdo al sitio en el que se encuentre un conjunto de tiendas, las cuales forman sectores a visitar.

Otro punto importante es el número de tiendas que visita cada vehículo, que en el caso de la empresa COLÁCTEOS, cada vehículo en promedio visita entre 150 y 200 tiendas, sin embargo las tiendas están ubicadas con distancias muy pequeñas entre sí, por lo cual se puede garantizar que los recorridos son eficientes con únicamente el cálculo del recorrido por sectores, es decir que no se encontrara el valor de distancia entre todas las tiendas, sino el de los sectores, puesto que para los 24 vehículos de la empresa el número de tiendas es de aproximadamente 5000, siendo un numero demasiado grande para el cálculo de la matriz de distancias e imposible computacionalmente.

Considerando entonces que el número de sectores incluido el deposito desde donde los vehículos se abastecen, salen a cubrir las rutas y regresan a entregar los recipientes junto con la orden de pedidos para el siguiente día. Se tiene 154 sectores incluido el depósito, para los 24 vehículos, entonces se realiza el cálculo de distancias entre cada uno de los sectores y se encuentra la matriz de costos de 154 filas x 154 columnas.

La importancia de encontrar una matriz de distancias entre sectores, radica en el acercamiento a la solución real del problema presente en la empresa, por ello es fundamental conocer el costo de ir desde cualquier origen a cualquier destino, para así aplicar el algoritmo genético en busca del conjunto de rutas óptimas para cada vehículo de la empresa y generar ahorros en combustible, tiempo y el aporte de brindar un servicio eficiente a la comunidad.

La secuencia para la programación y búsqueda de la matriz de costos se presenta con mayor detalle en el *anexo 9.5*

5 RESTRICCIÓN DE DEPENDENCIA DEL TIEMPO PARA LAS RUTAS

5.1 TOMA DE DATOS Y CÁLCULO DE DENSIDAD VEHICULAR

En la restricción de dependencia del tiempo o el horario de salida de los vehículos se realiza aforos vehiculares en diferentes puntos de la ciudad de San Juan de Pasto, esto con el fin de encontrar el valor de densidad vehicular en las diferentes calles de la ciudad.

Para esto se realiza aforos el día sábado con el fin de tomar una medida sin restricción vehicular, el día domingo con el fin de tomar una medida sin restricción y feriado. Como indicador de un día común y corriente que tiene restricción vehicular tomamos un día cualquiera, en este caso se toma valores el día jueves. Todos los datos se toman en las mismas horas y las mismas calles haciendo referencia a que se espera que la congestión en las diferentes vías sea alta, media y baja.

Las horas en las cuales se realiza el aforo son desde las 7:00 am hasta las 9:00 am, luego desde las 12:00 a las 2:00 pm y finalmente a las 7:00, 8:00 y 9:00 pm. Siendo estas las horas pico en las que la congestión vehicular es mayor.

Los resultados de estos datos se corren como un diseño experimental de 3 niveles y 3 factores así:

Tabla 5-1. Combinación del diseño factorial

Factor	Niveles
Hora	Mañana
	Tarde
	Noche
Día	Sábado
	Domingo
	Jueves
Categoría	Alta
	Media
	Baja

Como replicas se toma el valor de densidad vehicular de cada hora y cada día del experimento, los tres tipos de calle y a las diferentes horas consideradas pico en la ciudad de San Juan de Pasto.

5.2 ANÁLISIS ESTADÍSTICO.

Con los datos de densidad se corre el diseño experimental así:

Usando el software SPSS, se toman las variables como ordinales para cada nivel.

Luego se genera la tabla de datos con los tres niveles enumerados como 1, 2, 3 para cada variable, donde la variable *Día* representa en los valores de 1, 2 y 3 los días *Jueves*, *Sábado* y *domingo* respectivamente. En la variable *Hora* el valor de 1 significa *mañana*, 2 corresponde a *medio día* y 3 a *noche*. La variable categoría representa al tipo de calle 1 indica congestión *alta*, 2 *media* y 3 *baja*.

El proceso de análisis de varianza se realiza de acuerdo a las respuestas de los datos usando un nivel de confianza del 95%. Siguiendo los siguientes pasos:

1. Se realizan pruebas para verificación de supuestos de homogeneidad de varianzas mediante la prueba de Levene, las pruebas de normalidad de los residuos y los datos usando la prueba estadística de Kolmogorov Smirnov con la corrección de Lilliefors y la prueba de Shapiro- Wilk.
2. En caso de no cumplirse los supuestos, se recurre a realizar el análisis de varianza por medio de pruebas no paramétricas (Kruskal Wallis o Friedman), o por medio de la generación de rangos como lo recomienda Montgomery[57].
3. Se realiza un análisis de comparaciones múltiples en caso de existir diferencia significativa entre los tratamientos, para ello se usa las pruebas estadísticas de DMS (Diferencia mínima significativa) y se corrobora con la prueba de Tukey.
4. Se observa las gráficas de perfil con el fin de encontrar la existencia o no de interacción entre los factores.
5. De acuerdo con los resultados dados por en ANOVA, como en la prueba de rangos, se encuentra diferencias por no cumplimiento de supuestos, por lo que se realiza un diseño factorial completamente al azar para el factor HORA de forma independiente.

Los resultados del análisis muestran que no existe diferencia en las horas de la densidad vehicular, por lo que no es necesario hacer comparaciones múltiples.

La respuesta de la comparación de días, indica que el día de semana jueves es igual estadísticamente al día sábado. El día domingo presenta diferencia entre los días jueves y sábado, aunque no influye en la dependencia del horario de las rutas puesto que este día no generan servicios en la ciudad.

Por otra parte teniendo en cuenta los resultados del ANOVA se encuentra que existe diferencia significativa para los diferentes tipos de calles con las categorías alta, media y baja.

Con esto se determina realizar una diferencia de medias para variables con muestras distintas y con varianza distinta, haciendo uso del toolbox de estadísticas de matlab y la prueba T debido a la poca cantidad de datos de densidad que se tiene para las distintas calles de la ciudad de San Juan de Pasto.

De acuerdo con la información de diferencia de medias entre las diferentes calles muestreadas, se tiene una clasificación ALTA, MEDIA y BAJA, lo cual permite generar un costo al recorrido de acuerdo al conocimiento del conductor en la ciudad, es decir que según la ruta que se genere se hace un incremento del costo dado por la vía que se transita considerando a esta con un valor según la categoría mencionada. (Ver anexo 9.4)

6 PRUEBAS Y RESULTADOS OBTENIDOS

Para la comprobación del algoritmo genético en el problema de ruteo de vehículos se generan varias corridas de la programación desde el software matlab instalado en un computador portátil con un procesador Intel Core2 dúo y 2 Gb de memoria RAM. Con dichas características se realiza las corridas para diferentes Instancias conocidas, con el propósito de validar el algoritmo y poderlo aplicar en el caso real de interés. Las referencias tomadas se encuentran en la literatura como instancias de AUGERAT, considerando desde 32 hasta 80 clientes, de los cuales se conoce el valor óptimo.

Como prueba se hace 10 corridas por cada instancia y se obtiene el valor de la función objetivo, el error con referencia al óptimo, la ruta de cada vehículo y el grafico de las rutas.

A continuación se presenta la tabla 6-1, la cual indica el valor de error alcanzado en las diferentes corridas de algunas instancias.

El nombre de la instancia se caracteriza por llevar el número de ciudades y el número de rutas, así como ejemplo la instancia AUGERAT39K5 indica un problema de ruteo para 39 clientes y 5 rutas.

Tabla 6-1. Resultados para las corridas del algoritmo genético para diferentes instancias.

Instancia	Valor óptimo	Función objetivo mínima encontrada	% error mínimo	Tiempo de ejecución seg.
Augerat32k5	784	803,27	2,46	245
Augerat33k6	742	751,86	1,33	149
Augerat34k5	778	789,9	1,53	252
Augerat37k6	949	982,330	3,512	323
Augerat38k5	730	737,56	1,04	303
Augerat39k5	822	848,972	3,28	311
Augerat46k7	914	936	2,14	514
Augerat60k9	1408	1401	-0,4	391
Augerat80k10	1764	1839,1	4,26	616

Los valores promedio de porcentaje de error mínimo en todas las pruebas son de 4,73% y el tiempo promedio de respuesta para todas las instancias es de 415 segundos.

La respuesta grafica al correr el algoritmo en matlab se indica así:

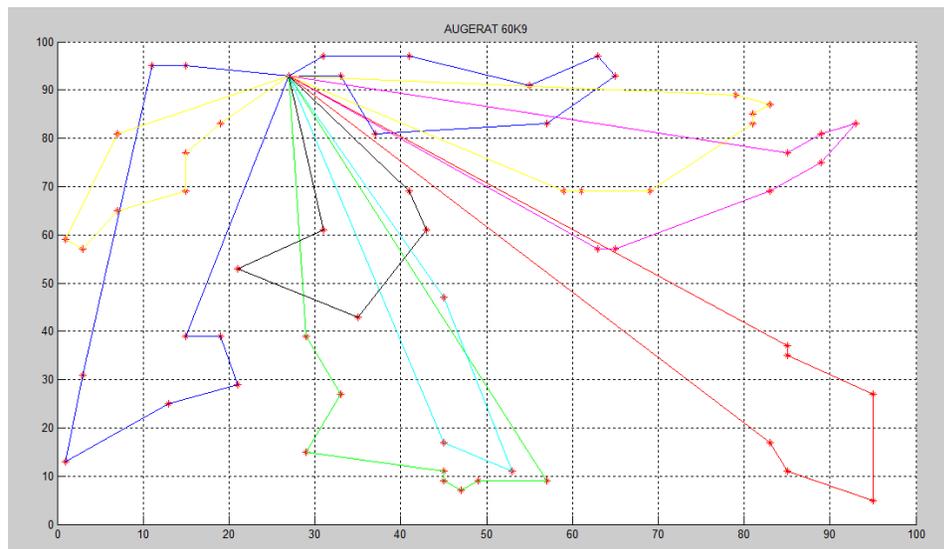


Figura 6-1. Respuesta para las rutas de la instancia Augerat60k9

En la figura 6-1 se observa una de las mejores respuestas para el problema de ruteo de vehículos con capacidad fija, cuya respuesta presenta un valor de error del -0,4% con respecto al valor óptimo, es decir que se mejora en 0.5% el valor de la instancia publicada por Augerat para 60 clientes con 9 rutas.

Tabla 6-2. Valor de la función objetivo para instancia de Augerat60k9

Mejor Solución	F. objetivo acumulada	error	Tiempo de respuesta
1401	1408	-0,496 %	399 seg.

En la tabla 6-2 se indica los valores de la respuesta de la instancia tanto para el valor óptimo como para la respuesta encontrada con el algoritmo genético, además del valor de error y el tiempo en que se encuentra la respuesta.

Tabla 6-3. Rutas de la solución a la instancia de Augerat60k9

RUTAS								
8	7	7	6	8	3	5	8	7

En la tabla 6-3 se encuentran las rutas de los vehículos, así el vehículo uno tendría que recorrer las 8 primeras ciudades, el segundo vehículo las 7 ciudades siguientes y así hasta recorrer el total de 59 ciudades cada uno saliendo y retornando al depósito.

La secuencia de las visitas se indica en el siguiente vector:

Tabla 6-4. Secuencia de visitas a las ciudades por cada vehículo.

Ruta1	19	15	20	60	39	53	34	42
Ruta2	36	56	16	9	14	30	8	
Ruta3	28	58	38	18	27	40	51	
Ruta4	44	57	13	52	10	33		
Ruta5	46	6	55	11	23	37	2	3
Ruta6	49	46	59					
Ruta7	35	7	25	24	48			
Ruta8	17	21	54	31	50	45	29	32
Ruta9	26	47	41	12	22	5	4	

De igual manera se hace el corrido del modelo obtenido para la matriz de costos de las distancias entre sectores en la ciudad de San Juan de Pasto, usando el mismo algoritmo genético, en el cual se genera una población de 20 soluciones al problema y se vuelve a correr el programa una vez más para obtener la respuesta de menor costo posible, a la cual en seguida se le aumenta el costo del tiempo a la matriz de costos y se genera la mejor respuesta al problema de ruteo de vehículos con dependencia de tiempo.

De acuerdo con lo anterior se obtiene los resultados indicados en la tabla 6-5:

Tabla 6-5. Respuestas de la función objetivo para 20 corridas del algoritmo genético en 153 sectores de la ciudad de San Juan de Pasto.

Valor de la función objetivo en Km	Tiempo de ejecución seg.	Número de rutas
466,301	1493,19009	18
458,587	880,146863	18
458,727	801,910077	18
459,245	787,392897	18
457,787	759,051592	18
450,734	761,178962	18

455,001	788,243847	18	
459,216	755,043834	18	
460,435	762,629782	18	
458,958	758,74785	18	
460,58	738,747457	18	
453,098	729,173048	18	
461,148	786,727717	18	
453,98	732,86424	18	
457,8	735,993603	18	
460,543	731,082975	18	
458,073	739,831898	18	
458,285	745,702042	18	
460,34	736,460857	18	
455,956	743,925202	18	
Promedio	458,24	798,4	18
Desviación	3,35	167,259	0

Con respecto a las corridas del algoritmo genético de forma independiente para el caso de los *153* sectores a los cuales se debe llevar el producto desde la empresa Colácteos, se encuentra que se puede cubrir todos los sectores con *18* vehículos con capacidad fijada en *70* canastos.

El valor de la función objetivo encontrada presenta un valor promedio de *458,24* Km, los cuales los recorren los *18* vehículos. Además se observa que la desviación entre las *20* corridas es mínima con un valor de *3,35km*.

El tiempo promedio que gasta el computador es de *798,4* segundos en cada corrida.

Con estos valores se procede a realizar una última corrida del algoritmo genético, tomando como población inicial las *20* soluciones obtenidas anteriormente, esperando observar una mejora pequeña o al menos seleccionando la mejor de las soluciones anteriores.

En este caso se logra obtener como resultado de la función objetivo el valor de ***446,147 Km*** en un tiempo de ***1337,75 Segundos***.

Para este caso no es conveniente indicar una respuesta gráfica desde matlab, puesto que no se tiene el valor de las coordenadas cartesianas de cada uno de los sectores, sino que se hace uso de la matriz de costos obtenida anteriormente a partir de la consulta de *googlemaps* con coordenadas de latitud y longitud.

Para lograr visualizar las rutas obtenidas se introduce a la web en la API de googlemaps el valor de las coordenadas de latitud y longitud de acuerdo con la secuencia de cada ruta encontrada.

Tabla 6-6. Rutas para sectores en la ciudad de San Juan de Pasto.

Vehículo	Secuencia de sectores partiendo del depósito y terminando en el depósito
1	96 15 104 66 68 23 75 69 74 64 63 65 143
2	91 147 148 17 14 21 13 6
3	92 24 82 116 4 5 3 7
4	102 2 118 117 22 12 11 115 101
5	121 30 28 110 119 133 103
6	93 29 122 10 18 19 97 99 9 149 154
7	48 146 145 109 107 123
8	124 31 49 112 53 60 137 136 125
9	150 108 113 111 56 59 55
10	26 27 135 140
11	120 25 134 61 57 114 151
12	100 58 54 153 89 152 90 132
13	51 86 88 130 127 128
14	72 71 78 79 32 33 41 44 43 42 106 141 139
15	85 131 87 52 126 84 129
16	144 95 94 34 73 70 67 62 77 80 81
17	138 83 47 45 76 46 50 20 16 98 8 142
18	40 37 39 38 35 36 105

Como un valor comparativo del estado actual que presenta la empresa, se indica en la tabla 6-7, el valor de la distancia que recorre cada uno de los 24 vehículos y el valor promedio de costo por combustible de cada vehículo.

Tabla 6-7. Costos en distancia y combustible para rutas con las que actualmente cuenta la empresa.

Vehículo	Costo combustible CP\$	Distancia ruta Km
1	15000	22,3
2	15000	23,8
3	11000	15,6
4	17500	19
5	30000	44,5
6	24000	21,8

7	10000	20,8
8	15000	34,4
9	12000	17,9
10	15000	21,4
11	12000	16,4
12	15000	18,4
13	15000	24,8
14	15000	22,6
15	18000	22
16	15000	14
17	15000	23,4
18	15000	14,4
19	10000	10,3
20	15000	14,5
21	15000	15
22	17000	16,4
23	12000	17,4
24	30000	25,9
SUMA	383500	497

Como lo indica la tabla 6-7, el costo en distancia de las rutas es de 497 Km y el valor del combustible es de \$ 383.500. Es decir que la relación de costo de combustible con distancia es de $\$383500/497\text{Km}$, igual a $\$771,63/\text{km}$. Además que para cubrir todos los sectores se requiere de 24 vehículos.

Con la respuesta que da el algoritmo genético, se tiene únicamente 18 rutas que recorren un total de 446Km aproximadamente, valor que es mucho menor al valor que genera la aplicación de googlemaps, el cual genera el valor mínimo de costo al trazar la ruta de los sectores que los vehículos deben visitar, dicho valor es muy probable que en la realidad de los conductores sea mucho más alto puesto que ellos lo hacen con base en su experiencia y no en el trazado de la ruta según algoritmos internos de google.

El valor del combustible de la ruta nueva esta alrededor de $771,63\$\ast 446\text{Km}$, igual a $\$344147$. Valores que se pueden mejorar al introducir los algoritmos de googlemaps según la secuencia de visita de sectores planteada por la ejecución del algoritmo genético en matlab.

Para el caso de la secuencia generada por el algoritmo genético se indica en la figura 6-1, una de las 18 rutas geo referenciadas en la ciudad de San Juan de Pasto, junto con el valor de distancia en Kilómetros que recorrerá el vehículo.

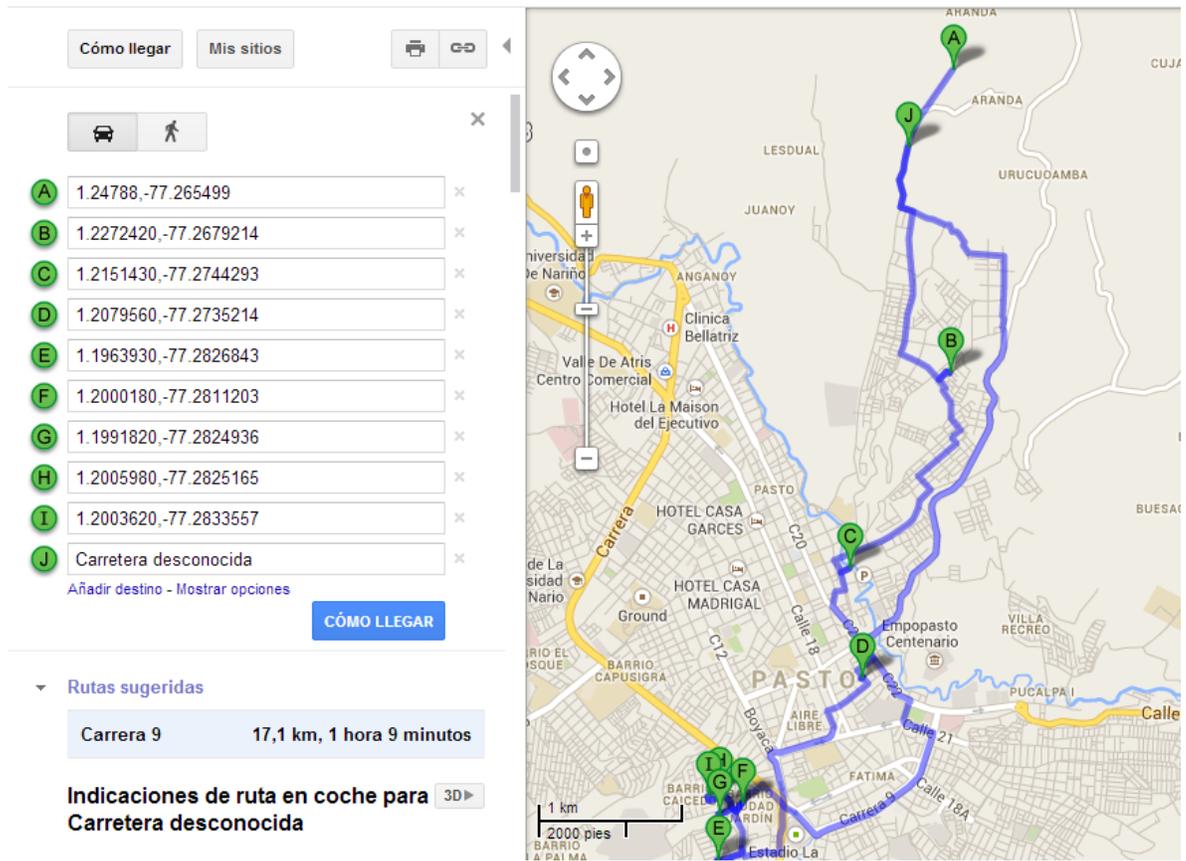


Figura 6-2. Ruta del vehículo 3.

En la figura 6-2, se ve la secuencia de coordenadas en latitud y longitud de la ruta número 3, con una distancia total recorrida de 17,1 Km. Siendo esta la distancia total de la ruta, la cual si se hace un cálculo sencillo de velocidad estaría en un valor medio de 14,9Km/h sin contar con la densidad vehicular.

En resumen se presenta la tabla 6-8 con los valores de distancia y tiempo recorrido por cada una de las rutas:

Tabla 6-8. Resumen de distancia y tiempo de recorrido de las 18 rutas para distribución de productos lácteos en la ciudad de San Juan de Pasto.

RUTA	Distancia Km	Tiempo Horas	Velocidad media en Km/h
1	20,6	1,55	13,29
2	17,3	1,13	15,26
3	17,1	1,15	14,87
4	20,2	1,40	14,43

5	14,3	1,22	11,75
6	17,8	1,35	13,19
7	15,5	1,25	12,40
8	16,9	1,32	12,84
9	17	1,38	12,29
10	11,7	0,90	13,00
11	15,8	1,22	12,99
12	19,2	1,25	15,36
13	16,2	1,12	14,51
14	35,1	2,03	17,26
15	18,6	1,22	15,29
16	20	1,42	14,12
17	18,5	1,35	13,70
18	16,3	1,15	14,17
SUMA	328,1	23,4	Promedio=13,9 3Km/h

Es importante notar como el valor total de distancia recorrida por las 18 rutas (costo) presenta un valor de *328,1Km*. Que comparado con el valor de referencia de 497 Km de la empresa COLÁCTEOS se tiene un ahorro de *168,9 Km*, los cuales representan un ahorro del *34%*, sin contar con que el valor de distancia de la empresa ya tiene un costo reducido con respecto a la realidad puesto que este se obtuvo desde la generación de rutas apoyado con googlemaps.

Tabla 6-9. Comparativo de rutas de Colácteos con Rutas propuestas

	No de rutas	Distancia total en Km	Costo combustible CP\$
Colácteos	24	497	383500
Algoritmos genéticos	18	328,1	253171,8
Diferencia porcentual	25%	34%	34%

Con respecto al valor aproximado de combustible se puede decir que el kilómetro recorrido tiene un costo de \$771,63, y multiplicado por los *328,1 Km* de las rutas es igual a \$253 172. Que comparado con el valor de referencia de la empresa de \$ 383.500, presenta un ahorro de \$ 130 328 diarios.

Para incluir la restricción del tiempo se genera una matriz de costos sumando un valor de distancia a las vías consideradas con densidad alta, y media, lo cual está dado de acuerdo con los valores encontrados en el análisis estadístico anterior, para lo cual se toma el valor de densidad vehicular en horas pico, con la que se toma el valor de velocidad de flujo

vehicular en las diferentes calles, dicha velocidad se encuentra a partir de la distancia recorrida por unidad de tiempo, por tanto se encuentra un valor de distancia que corresponda a un valor directamente proporcional a la densidad vehicular en horas pico, con lo cual se tendrá un problema de ruteo dependiente del tiempo y se resuelve igual que el ruteo de vehículos capacitado en horas pico y no pico.

De acuerdo con los datos de aforo vehicular, se resume en la tabla 6-9, los valores de velocidad media para cada calle aforada, junto con el valor de densidad vehicular y el número de vehículos totales.

El aforo para obtener densidad vehicular se hace por medio del conteo de vehículos en cada calle durante 15 minutos en una distancia fija de 30 metros. A cada vehículo se le toma el tiempo en que recorre la distancia para obtener el valor de velocidad, luego se toma el número de vehículos por hora y se divide entre el valor de velocidad media de todos los vehículos para obtener la densidad vehicular en unidades de Vehículos/Km.

Tabla 6-10. Velocidad media y crítica para calles clasificadas en categoría alta.

Clasificación	Ref Alta	alta	alta	alta	alta
Calle	Principal Lorenzo Alta	Cll 17 con AV américas	Carrera 32 entre calles 16 a y 16 b	Iglesia de San Sebastián (de la Panadería)	Las Margaritas
Densidad promedio Veh/km	62,867	76,95	77,16	63,16	47,91
Número de vehículos	7725	2416	444	1095	2392
Velocidad promedio según aforo Km/h	25,742	16	6,69	8,83	24,84
Velocidad crítica	6,83	7,85	1,44	4,33	12,48

Tabla 6-11. Velocidad media y crítica para calles clasificadas en categoría media.

Clasificación	Ref Media	media	media	media	media	media	media	media	media	media
Calle	calle 15 Media	Calle 15 Cr 23 y 24	Calle 15 Cr 26 y 27	Calle 15 Cr 25 y 26	Cr 22 Cll 16 y 17	Carrera 32 entre calles 15 y 16	Carrera 32 entre calles 18 y 19	Colegio Javeriano entre carreras 24 y 25	Hospital Infantil	Calle 15 Cr 22 y 23
Densidad promedio Veh/km	30,07	50,39	34,32	31,62	19,05	43,09	34,01	21,37	33,50	32,01
Número de vehículos	3316	792	657	604	923	678	672	1047	1592	455
Velocidad promedio según aforo Km/h	25,72	15,56	19,57	19	24,77	20,97	19,76	25,87	23,64	18,76
Velocidad crítica	6,13	3,93	4,79	4,78	12,11	3,93	4,94	12,25	11,88	3,55

Tabla 6-12. Velocidad media y crítica para calles clasificadas en categoría baja.

Clasificación	Ref Baja	baja	baja	baja	baja
Calle	Capusigra Baja	CAI obrero	ClI 18 Av chile	Cr 24 con dos puentes	Barrio Capusigra (entrada al Éxito)
Densidad promedio Veh/km	3,41	4,69	14,04	9,52	5,41
Número de vehículos	555	206	585	312	239
Velocidad promedio según aforo Km/h	31,36	21,612	20,94	16,48	23,1
Velocidad crítica	9,04	10,97	10,42	8,20	11,04

De acuerdo con los valores de densidad y el número de vehículos contados cada 15 minutos se obtiene la velocidad media de los aforos y la velocidad crítica, la cual está dada por el valor de la cantidad de vehículos totales sobre el número de periodos en que se divide la hora y sobre la densidad vehicular.

Así por ejemplo la **calle 17 con AV américas**. Tiene un valor de densidad vehicular de **62,867 Vehículos/Km**. y un total de vehículos aforados de 2416 tomados en 15 minutos, es decir que la velocidad crítica es igual a:

$$Velocidad\ crítica = \frac{\left(\frac{2416\ veh\acute{u}culos}{4\ hora}\right)}{\frac{76,95\ veh\acute{u}culos}{Km}} = \frac{7,85Km}{h}$$

En el caso de las calles de referencia se debe tomar el número de vehículos y dividirlos entre 18, puesto que es la cantidad de datos tomada para la densidad vehicular en dos días con 9 valores de densidad promedio en cada día distribuidos en tres horas en la mañana, tres horas al medio día y tres horas en la noche. Por lo tanto el número de vehículos totales se debe dividir entre 18 horas en las cuales se tiene el aforo. Así por ejemplo para la calle de referencia alta **Principal Lorenzo** se tiene:

$$Velocidad\ crítica = \frac{\left(\frac{7725\ veh\acute{u}culos}{18\ hora}\right)}{\frac{62,867\ veh\acute{u}culos}{Km}} = \frac{6,83Km}{h}$$

De acuerdo con lo anterior y la clasificación estadística hecha en capítulos anteriores se toma un valor promedio para el valor de velocidad crítica en cada categoría como se indica en la tabla 6-12:

Tabla 6-13. Valores de velocidad media por cada categoría.

Velocidad promedio categoría alta	6,14 Km/h
Velocidad promedio categoría media	7,15Km/h
Velocidad promedio categoría baja	9,93Km/h

La clasificación y el valor promedio indican como en calles de categoría alta, la velocidad es menor que en calles con categoría media y calles con categoría baja, indicando así que la categoría está dada de acuerdo a la congestión vehicular.

Finalmente se puede tomar como referencia el valor de velocidad media que sugiere la aplicación de googlemaps, que generalmente usa un valor de velocidad para los vehículos de **13,93km/h**.

Haciendo una relación entre la velocidad calculada por googlemaps y los valores de velocidad crítica promedio se tiene:

Tabla 6-14. Relación de velocidad media con velocidad crítica por categoría.

Velocidad media según googlemaps Km/h	Velocidad media por cada categoría Km/h	Factor de relación por categoría
13,93	6,14	2,27
13,93	7,15	1,95
13,93	9,93	1,4

Usando el valor de la relación de velocidad se genera un costo adicional en la matriz de distancias para las calles consideradas de categoría alta y media, asignando al valor de distancia calculada entre puntos un nuevo costo dado por la multiplicación del factor encontrado.

Para esto se identifican las rutas que se afectan de forma importante por las calles con mayor congestión, para así asignarles el costo extra de acuerdo con la relación anterior, usando la respuesta de googlemaps.

Según las rutas obtenidas con googlemaps, se realiza la identificación de las calles de alta y media densidad usadas por cada ruta y los sectores a los cuales se dirige el vehículo de la ruta, para así asignar el sobrecosto en la matriz de distancias de 2,27 en calles consideradas de categoría o congestión alta y 1,95 en calles de congestión media.

Luego se establece el costo de las rutas con las medidas reales de distancia entre sectores más el factor de sobre costo por congestión vehicular en las horas pico. Por ejemplo en la tabla 6-15 se indica los valores de sobre costo total para la ruta 1.

Tabla 6-15. Costos de ruta 1, incluyendo la congestión vehicular en horas pico

RUTA	Correspondencia del sector de cada ruta en matriz de costos según coordenadas (LAT LONG) para los 153 sectores y el depósito	Factor de sobre costo	Costo en distancia entre sectores Km	Distancia afectada por sobre costo para los sectores congestionados Km.
1	1 a 96	7,88	8,863	69,84044
	96 a 15	6,49	1,179	7,65171
	104 a 66	1,95	0,438	0,8541
	75 a 69	1,95	0,104	0,2028
	69 a 74	2,27	1,772	4,02244
	74 a 64	1,95	2,322	4,5279
	64 a 63	1,95	0,282	0,5499
	63 a 65	1,95	0,139	0,27105
	65 a 143	3,9	3,324	12,9636

Con base en el costo calculado en las calles de congestión alta y media se pasa a calcular el costo total de las rutas, esto se realiza con el costo obtenido en cada ruta según googlemaps, más el costo afectado por el factor de sobre costo y menos el costo de las calles afectadas puesto que este ya se incluye en la distancia afectada por el sobre costo (Columna 3 de la tabla 6-15).

Tabla 6-16. Costo de cada ruta y total afectado por sobre costo de tráfico en hora pico

RUTA	Distancia de cada ruta en Km	Distancia Km afectada por factor de sobre costo
1	20,6	103,06094
2	17,3	143,19096
3	17,1	51,20631
4	20,2	48,53615
5	14,3	66,35419
6	17,8	52,19795
7	15,5	16,13373

8	16,9	38,97283
9	17	62,8407
10	11,7	53,82202
11	15,8	50,70584
12	19,2	123,15686
13	16,2	62,59618
14	35,1	111,97748
15	18,6	72,66788
16	20	61,58298
17	18,5	34,52001
18	16,3	16,3
SUMA	328,1	1169,82
RELACIÓN		3,565

Según la tabla 6-16 se observa el costo de cada ruta según el factor de sobre costo dado por la congestión vehicular en horas pico. Como resultado final se tiene que la congestión vehicular afecta en 3,57 veces al costo total de las rutas.

Un elemento importante para este tipo de problemas en los que interviene el horario en que se generan las rutas es la comparación de resultados específicos con instancias propuestas en la literatura especializada, sin embargo para el caso particular los autores que han dado soluciones al problema del TDVRP no presentan instancias específicas a este problema y para validar los resultados obtenidos se recurre a comparación con instancias publicadas para problemas semejantes como el CVRP, o el VRPTW para el cual existe gran número de instancias publicadas por Solomon como se indica en el artículo de Duygo, Dellaert, et al. [19].

En el caso particular desarrollado para la empresa COLACTEOS, se realiza la comparación de las respuestas obtenidas por el algoritmo genético y las desarrolladas con las instancias publicadas por Augerat para el problema del CVRP con resultados cercanos a los óptimos.

7 CONCLUSIONES Y RECOMENDACIONES

1. De acuerdo con la revisión del estado del arte y la búsqueda de artículos en la literatura especializada se encuentra que el problema de ruteo de vehículos tiene una gran variedad de restricciones que acercan al problema de la realidad aunque la solución a este tipo de problemas combinatoriales se dificulta en la medida en que se tengan mayor cantidad de clientes. Dado esto en problemas en los que la realidad limita al desarrollo de métodos exactos para la solución, se generan diferentes métodos, tanto heurísticos como metaheurísticos que permiten obtener respuestas adecuadas para la implementación a la realidad.
2. En el desarrollo del proyecto se establece la metodología del algoritmo genético modificado como una metaheurística para la solución del problema, para lo cual se implementa el código en el software matlab, el que por su gran capacidad de manejo matemático por funciones, facilita en gran medida la operatividad de la codificación algorítmica.
3. Se propuso crear para el algoritmo genético modificado una población inicial híbrida que considera soluciones encontradas a través de técnicas heurísticas y soluciones generadas de forma aleatoria con un desempeño basado en una estructura de búsqueda aleatoria robusta que permite ajustar diferentes métodos tanto exactos como heurísticos y metaheurísticos en el mejoramiento de las soluciones, dando a esta técnica la flexibilidad suficiente en su codificación.
4. Se planteó una metodología por medio del estudio estadístico de la medida de densidad vehicular en la malla vial de algunas vías de la ciudad de San Juan de Pasto.
5. Se resolvió el problema de ruteo de vehículos con la restricción de dependencia del tiempo u horario de salida de los vehículos, considerando distintos escenarios basados en la congestión vehicular en horas pico en los días no feriados. Su implementación se realizó mediante el algoritmo genético modificado de Chu – Beasley.
6. El proceso de mutación en un algoritmo genético permite refrescar las soluciones de la población que tras un conjunto de varias iteraciones pueden tender a encontrar una solución óptima local lo que conlleva a un desgaste computacional, el cual se evita al introducir a la población un hijo con características diferentes a la tendencia de la solución y poder continuar por medio de combinación con la búsqueda de posibles mejores soluciones.
7. La efectividad de los resultados del algoritmo genético también radica en el uso del proceso de selección por torneo, permitiendo tomar al hijo más competitivo del algoritmo e introducirlo a la población.

8. Dada la flexibilidad en la codificación del algoritmo genético, en el problema se opta por realizar un mejoramiento de la solución final por medio de intercambios de aristas en las rutas finales y la implementación de rutinas de solución básicas como la técnica de ahorros y algoritmo del vecino más cercano de Dijkstra, lo cual permite obtener errores en promedio del 0,5% en las instancias publicadas para el CVRP.
9. El uso del aplicativo de googlemaps permitió tener un acercamiento a la realidad de las rutas en la ciudad de San Juan de Pasto para los vehículos de la empresa de distribución de productos lácteos COLACTEOS. Dicha aplicación permite por medio de la georeferenciación de sectores en la ciudad tomar los valores de coordenadas en latitud y longitud de cada sector de interés y con base en dichas coordenadas se encuentra los valores de distancias entre sectores de acuerdo con las codificaciones de calles y carreras en la ciudad, estableciendo así la matriz de costos real.
10. Con el fin de estimar un costo de las rutas en la empresa, la aplicación de googlemaps permite obtener valores de distancia y velocidad media de los diferentes trayectos, con lo que se referencia el estado actual de las rutas de la empresa y su costo, que finalmente se compara con la solución obtenida por medio de los algoritmos genéticos que en el caso real generan un ahorro del 34%.
11. Los aforos vehiculares en las diferentes calles de la ciudad permiten tener una referencia del comportamiento de la densidad vehicular y por ende la congestión, que es un elemento importante en la descripción del comportamiento de las vías en distintas horas del día.
12. De acuerdo con las pruebas estadísticas y el análisis de varianza de los datos de densidad vehicular se establece que no existe diferencia significativa en la congestión vehicular en horas pico y entre los días de la semana excepto el día domingo en el que no se realiza recorrido de las rutas por parte de la empresa.
13. Por medio de los resultados de análisis de varianza de los datos y pruebas comparativas de diferencia de medias se establece la clasificación de las calles congestionadas como alta, media y baja; permitiendo así determinar el costo de las vías en función del tiempo de recorrido.
14. De acuerdo con los datos de velocidad media tomada de los aforos vehiculares y la velocidad media de googlemaps se establece una relación entre ellas para referir un sobrecosto de acuerdo con la congestión vehicular en las calles de la ciudad, con base en esto se generan las rutas para cubrir los 153 sectores con salida y retorno al depósito, obteniendo como resultado un sobrecosto por la dependencia del tiempo de 3,57 veces el costo de las rutas en horas no pico.
15. Se recomienda el uso de aplicaciones de ruteo de vehículos por medio de GPS que generen mayor información como congestión vehicular y alternativas de vías menos congestionadas en tiempo real.

16. A pesar que la programación en HTML y Java es adecuada en el uso de servicios de googlemaps, se puede incursionar en el uso de matlab y su toolbox googleearth_matlab.

8 REFERENCIAS BIBLIOGRÁFICAS

- [1] Y. V. Piqueras, "Optimización heurística económica aplicada a las redes de transporte del tipo VRPTW," Ph.D. Tesis. Escuela Técnica Superior de Ingenieros de Caminos, Canales y Puertos Universidad Politécnica de Valencia, May., 2002.
- [2] M. Sipser, *Introduction to the Theory of Computation.*, 3rd ed. Boston, USA., 2012, p. 448.
- [3] B. M. Baker and M. a. Ayechev, "A genetic algorithm for the vehicle routing problem," *Comput. Oper. Res.*, vol. 30, no. 5, pp. 787–800, Apr., 2003.
- [4] L. B. Rocha Medina, E. C. González La Rota, and J. A. Orjuela Castro, "Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y métodos de solución State of the art review of the vehicle routing problem: A historic account with solving methods Resumen," *Ing. Univ. Fr. José Caldas*, vol. 16, no. 2, pp. 35–55, Oct., 2011.
- [5] C. Malandraki and M. Daskin, "Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms," *Transp. Sci.*, vol. 26, no. 3, pp. 185–200, Aug., 1992.
- [6] A. V. Donati, R. Montemanni, N. Casagrande, A. E. Rizzoli, and L. M. Gambardella, "Time dependent vehicle routing problem with a multi ant colony system," *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1174–1191, Mar., 2008.
- [7] A. L. Kok, E. W. Hans, and J. M. J. Schutten, "Vehicle routing under time-dependent travel times: The impact of congestion avoidance," *Comput. Oper. Res.*, vol. 39, no. 5, pp. 910–918, May., 2012.
- [8] H.-K. Chen, C.-F. Hsueh, and M.-S. Chang, "The real-time time-dependent vehicle routing problem," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 42, no. 5, pp. 383–408, Sep., 2006.
- [9] M. J. E. Palacios, "Una formulación para el problema de ruteo de vehículos con tiempos de viaje dependientes del tiempo para la actualización de rutas con información en tiempo real". M.S Tesis. Pontificia universidad Católica de Chile, Escuela de ingeniería, Jul., 2009.

- [10] M. R. Khouadjia, B. Sarasola, E. Alba, L. Jourdan, and E.-G. Talbi, "A comparative study between dynamic adapted PSO and VNS for the vehicle routing problem with dynamic requests," *Appl. Soft Comput.*, vol. 12, no. 4, pp. 1426–1439, Apr., 2012.
- [11] N. Mladenović, D. Urošević, S. Hanafi, and A. Ilić, "A general variable neighborhood search for the one-commodity pickup-and-delivery travelling salesman problem," *Eur. J. Oper. Res.*, vol. 220, no. 1, pp. 270–285, Jul., 2012.
- [12] S. Kritzinger, K. F. Doerner, R. F. Hartl, G. Ľ. Kiechle, H. Stadler, and S. S. Manohar, "Using Traffic Information for Time-Dependent Vehicle Routing," *Procedia - Soc. Behav. Sci.*, vol. 39, pp. 217–229, Jan., 2012.
- [13] K. Fleszar, I. H. Osman, and K. S. Hindi, "A variable neighbourhood search algorithm for the open vehicle routing problem," *Eur. J. Oper. Res.*, vol. 195, no. 3, pp. 803–809, Jun., 2009.
- [14] J. F. Delgado O., "Modelo matematico para el analisis y mejoramiento de los metodos aproximados para la solucion de los problemas de enrutamiento, asignacion de horarios y distribucion de cargas: una aplicacion a la entrega de productos embotellados," Instituto tecnológico de CD. Madero, TAM. Mexico, Dic., 2007.
- [15] A. Imran, S. Salhi, and N. a. Wassan, "A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem," *Eur. J. Oper. Res.*, vol. 197, no. 2, pp. 509–518, Sep., 2009.
- [16] E. Stegers, "A Solution Method for Vehicle Routing Problems with Time-Dependent Travel Times," Delft University of Technology. Faculty of Electrical Engineering., Dec., 2009.
- [17] A. Subramanian, "Heuristic, Exact and Hybrid Approaches for Vehicle Routing Problems," Ph.D Thesis. UNIVERSIDADE FEDERAL FLUMINENSE, 2012.
- [18] A. Franceschetti, D. Honhon, T. Van Woensel, T. Bektas, and G. Laporte, "The Time-Dependent Pollution-Routing Problem," *Beta Res. Sch. Oper. Manag. Logist.*, vol. 409, no. 804, pp. 1–52, Feb., 2013.
- [19] D. Tas, N. Dellaert, T. Van Woensel, and T. De Kok, "The Time-Dependent Vehicle Routing Problem with Soft Time Windows and Stochastic Travel Times The Time-

Dependent Vehicle Routing Problem with Soft Time Windows and Stochastic Travel Times,” vol. 413, no. March., 2013.

- [20] S. R. Balseiro, I. Loiseau, and J. Ramonet, “An Ant Colony algorithm hybridized with insertion heuristics for the Time Dependent Vehicle Routing Problem with Time Windows,” *Comput. Oper. Res.*, vol. 38, no. 6, pp. 954–966, Jun., 2011.
- [21] A. Haghani and S. Jung, “A dynamic vehicle routing problem with time-dependent travel times,” *Comput. Oper. Res.*, vol. 32, no. 11, pp. 2959–2986, Nov., 2005.
- [22] Y. Kuo, C.-C. Wang, and P.-Y. Chuang, “Optimizing goods assignment and the vehicle routing problem with time-dependent travel speeds,” *Comput. Ind. Eng.*, vol. 57, no. 4, pp. 1385–1392, Nov., 2009.
- [23] E. Alba and B. Dorronsoro, “Computing nine new best-so-far solutions for Capacitated VRP with a cellular Genetic Algorithm,” *Inf. Process. Lett.*, vol. 98, no. 6, pp. 225–230, Jun., 2006.
- [24] S.-T. Bae, H. S. Hwang, G.-S. Cho, and M.-J. Goan, “Integrated GA-VRP solver for multi-depot system,” *Comput. Ind. Eng.*, vol. 53, no. 2, pp. 233–240, Sep., 2007.
- [25] Y. Marinakis and M. Marinaki, “Expert Systems with Applications A hybrid genetic – Particle Swarm Optimization Algorithm for the vehicle routing problem,” *Expert Syst. Appl.*, vol. 37, no. 2, pp. 1446–1455, Jun., 2010.
- [26] G. Jeon, H. R. Leep, and J. Y. Shim, “A vehicle routing problem solved by using a hybrid genetic algorithm,” *Comput. Ind. Eng.*, vol. 53, no. 4, pp. 680–692, Nov., 2007.
- [27] A. S. Tasan and M. Gen, “A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries,” *Comput. Ind. Eng.*, vol. 62, no. 3, pp. 755–761, Apr., 2012.
- [28] P. D. Ricardo Alberto Hincapié Isaza, Carlos Alberto Ríos Porras, Ramón Alfonso Gallego R, “modelo TSP,” *Sci. Tech. No 24.*, p. 6. Mar., 2004.
- [29] J. AHUJA, R.; MAGNANTI, T. and ORLIN, *Network flows: theory, algorithms, and applications.*, 1. ed. Prentice Hall. 1993.

- [30] A. OLIVERA, "Heurísticas para problemas de ruteo de vehículos," *Inst. Comput. Fac. Ing. Univ. la República, Montevideo, Uruguay.*, Agt., 2004.
- [31] R. H. BALLOU, *Logística empresarial. Control y planificación*, Edición 1a. , p. 655. Madrid, 1991.
- [32] H. D. BAZARAA, M.S.; JARVIS, J.J.; SHERALI, *Programación lineal y flujo en redes.*, Ed. Limusa. 1998.
- [33] R. E. GOMORY, "Outline of an algorithm for integer solutions to linear programs.," *Bull. Am. Math. Soc.*, vol. 64, pp. 275–278., 1958.
- [34] N. Azi, M. Gendreau, and J.-Y. Potvin, "An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles," *Eur. J. Oper. Res.*, vol. 202, no. 3, pp. 756–763, May., 2010.
- [35] W. R. GOLDEN, B.L.; STEWART, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization.* pp. 207–249., 1985.
- [36] M. M. KOSKOSIDIS, Y.A.; POWELL, W.B.; SOLOMON, "An OptimizationBased Heuristic for Vehicle Routing and Scheduling with Soft Time Window Constraints.," *Transp. Sci.* 26(2)69-85., vol. 31, no. 2, pp. 170–186., 1997.
- [37] H. W. J. M. KOLEN, A.W.J.; RINNOOY KAN, A.H.G.; TRIENEKENS, "Vehicle Routing with Time Windows.," *Oper. Res.* 35 266-273., vol. 35, no. 2, p. 266 a 273, 1987.
- [38] A Escobar, R. Gallego, E, Toro., *Técnicas metaheurísticas de optimización*, Segunda ed.Pereira, Universidad Tecnológica de Pereira, 2008.pp. 315.
- [39] J. R. BAKER, E.K.; SCHAFFER, "Solution improvement heuristics for the vehicle routing and scheduling problem with time window constraints.," *Am. J. Math. Manag. Sci.*, vol. 6, no. 3–4, pp. 261–300., 2013.
- [40] W. CLARKE, G.; WRIGHT, "Scheduling of vehicles from a central depot to a number of delivery points.," *Oper. Res.* 12 568-581., vol. 66, no. 3, pp. 568–581, Dic., 1993.
- [41] J. M. POTVIN, J.Y.; ROUSSEAU, "A parallel route building algorithm for the vehicle routing and scheduling problem with time windows.," *Eur. J. Oper. Res.* 66 331-340., vol. 66, no. 3, pp. 331–340, Jul., 1993.

- [42] J. A. And and D. Ulrich., "A New Parallel Tour Construction Algorithm for the Vehicle Routing Problem with Time Windows," *Univ. Zu Loin, Lehrstuhl Fur Wirtscha*, p. 21, Mar., 1995.
- [43] J. F. KONTORAVDIS, G.A.; BARD, "A GRASP for the Vehicle Routing Problem with Time Windows.," *Arrowsmith Technol. Inc., Austin, TX 78705*, pp. 10–23, 1995.
- [44] O. BRÄYSY, "Five Local Search Algorithms for the Vehicle Routing Problem with Time Windows.," *Transp. Sci.*, vol. 39, no. 1, pp. 104–118, 2005.
- [45] J. CHRISTOFIDES, N.; BEASLEY, "The period routing problem.," *Networks 142*, 237–246., vol. 14, no. 2, pp. 237–256, 1984.
- [46] H. N. THOMPSON, P.M. PSARAFTIS, "Cyclic transfer algorithms for multivehicle routing and scheduling problems.," *Oper. Res.*, vol. 41, no. 5, pp. 935–946, Sep-Oct., 1993.
- [47] T. SCHULZE, J.; FAHLE, "A parallel algorithm for the vehicle routing problem with time window constraints.," *Ann. Oper. Res.*, vol. 86, no. 0, pp. 585–607, Feb., 1999.
- [48] J. P. OSMAN, I.H. KELLY, "Meta-Heuristics: Theory & Applications.," Kluwer Aca. Springer; Softcover reprint of the original 1st ed. 1996 edition Sept 26, Ed. 2011, pp. 690.
- [49] C. A. Martinez, "Metahéurísticas híbridas aplicadas al problema de ruteo de arcos capacitados," Ph.D. Tesis. Universidad de Buenos Aires - Universidad Nacional de Salta., 2011.
- [50] P. SHAW, "A New Local Search Algorithm Providing High Quality Solutions to Vehicle Routing Problems.," *Work. Pap. Univ. Strat. Glas. Scotland*. Dec., 1997.
- [51] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," *Eur. Conf. Artif. LIFE*, no. D, pp. 134–142., 1991.
- [52] G. GAMBARDELLA, L.M.; TAILLARD, É.; AGAZZI, "MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. In: Corne, D. et al.," *New Ideas Optim. McGrawHill. London.*, vol. 2, no. 0, pp. 63–76, Jun., 1999.
- [53] J. H. HOLLAND, *Adaptation in natural and artificial systems.*, Oxford, En. Michigan, 1992, p. 211.

- [54] T. Ralphs. (2003, Oct.). "Vehicle Routing Data Sets." [Online]. Available: <http://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/index.htm.old>.
- [55] Colacteos. (2009, Ene.) "Historia de la empresa de productos lácteos de Nariño" [Online]. Available: <http://www.colacteos.com/historia.html>
- [56] D. Montgomery, *Diseño y análisis de experimentos*, Segunda edición. New York, Ed. Limusa. 2004, pp. 692.
- [57] M. A. Figliozzi, "Planning approximations to the average length of vehicle routing problems with time window constraints," *Transp. Res. Part B*, vol. 43, no. 4, pp. 438–447, Aug., 2009.

9 ANEXOS

9.1 PROGRAMA PRINCIPAL PARA ALGORITMO GENÉTICO

```
clc
clear all
close all
tic
% Instancias para pruebas de VRP
inicio=augerat32()
%inicio=augerat80()
%inicio=augerat33k5()
%inicio=augerat33k6()
%inicio=augerat34k5()
%inicio=augerat36k5()
%inicio=augerat37k5()
%inicio=augerat37k6()
%inicio=augerat38k5()
%inicio=augerat39k5()
%inicio=augerat39k6()
%inicio=augerat44k7()
%inicio=augerat45k6()
%inicio=augerat45k7()
%inicio=augerat46k7()
%inicio=augerat48k7()
%inicio=augerat53k7()
%inicio=augerat54k7()
%inicio=augerat55k9()
%inicio=augerat60k9()
%inicio=augerat61k9()
%inicio=augerat62k8()
%inicio=augerat63k9()
%inicio=augerat63k10()
%inicio=augerat64k9()
%inicio=augerat65k9()
%inicio=augerat69k9()

% Estructura inicio con las variables de Distancia, Demanda, Capacidad, Número de
% ciudades y el valor óptimo.
D=inicio.D
Q=inicio.Q
Cap=inicio.Cap
nciudades=inicio.nciudades
MejorSolucion=inicio.MejorSolucion
% matriz de distancias
Dis=inicio.Dis;

ite=500;% Número de iteraciones para mejoramiento con 2OPT
padre1 = inf;
padre2 = inf;
Ipadre1=0;
Ipadre2=0;
%generaciones para iterar algoritmo genético de acuerdo al número de ciudades;
generaciones = ceil(nciudades*1000/80);
tasamutacion=0.05*generaciones; % 5% del número de generaciones
```

```

cm=1; % contador de mutaciones
cmh=1; % contador de hijos mutados, solo puede tener valores de 1 o 2
%para mutar un hijo cada vez.
% capacidad fija
TamPoblacion=30;
geninicial=1;Poblacion2=[];

% GENERAR POBLACIÓN INICIAL DE SOLUCIONES PARA AG QUE SE HARÁ ALEATORIAMENTE

for i=1:TamPoblacion
    Alternativa=randperm(nciudades);
    for m=1:nciudades
        if Alternativa(m)==1
            Alternativa(m)=nciudades+1;
            break
        end
    end
    Poblacion(i,:)=Alternativa;
% asignar la demanda de la ciudad correspondiente según el orden de la
%solución
    for k=1:nciudades
        PQ(i,k) = Q(Poblacion(i,k));
    end
end

% VECTOR DE RUTAS
ka=1;
cc=1;
sumador=0;
Pos1=0;
Pos2=0;
for ja=1:TamPoblacion
    % LLAMAR FUNCIÓN ASIGNAR CAMIONES
    CamionesPorIndividuo(ja,:)=FunCamVRP(PQ(ja,:),...
        Poblacion(ja,:),nciudades,Cap,ka,cc,sumador);
    % LLAMAR FUNCIÓN PARA EVALUAR LA RUTA
    cp(ja,:)=CamionesPorIndividuo(ja,:);
    ValFunObj(ja,1)=FunObjVRP(D,cp(ja,:),Poblacion(ja,:),nciudades);
    % SELECCIÓN DE INDIVIDUOS DE LA POBLACIÓN

    if (ValFunObj(ja))<= padre1
        padre2 = padre1;
        Ipadre2 = Ipadre1;
        Pos2=Pos1;
        padre1 = ValFunObj(ja);
        Pos1=ja;
        Ipadre1=Poblacion(ja,:);
    end

% si el primer valor de ValFunObj es mínimo, el padre2 es cero

if ja==TamPoblacion & padre2==inf
    ban=min(ValFunObj);
    banpos=ja;
    ValFunObj(ja)=inf;
    [padre2,Pos2]=min(ValFunObj);
end

```

```

    Ipadre2=Poblacion(Pos2,:);
    ValFunObj(banpos)=ban;
end

    C=0;
    camion=0;
    ObjPadres=[padre1,Pos1;padre2, Pos2];
end

    while geninicial<=2
% % LLAMAR POBLACIÓN DE LAS HEURÍSTICAS
% Esta función hace un llamado a los algoritmos básicos de solución de VRP para
% encontrar 20 soluciones de una población inicial actualizada.
    if geninicial==2 % llamar heurísticas
        Poblacion2=[];
        camiones2=[];
        fobj2=[];
        [Poblacion2,camiones2,fobj2]=heurísticas(nciudades,Cap,Q,D,ite,Dis);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% actualizar la población, camiones y función objetivo... y sacar a la
% población nueva los 10 mejores para luego adicionar los 20 individuos de
% las heurísticas

% llamar función para ordenar población y obtener los 10 mejores
Poblacion1=[];
camiones1=[];
fobj1=[];
[Poblacion1,camiones1,fobj1]=ordenar(Poblacion,cp,ValFunObj);

% guardo en una variable inicial las soluciones de la población mejoradas
RutaIniMejorada=[Poblacion1;Poblacion2];
CamionesIniMejorados=[camiones1;camiones2];
FOIniMejorada=[fobj1';fobj2'];
% valores de población, camiones y función obj actualizados para realizar
% genético con buenos individuos

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TSP para mejorar cada ruta de la población%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for t=1:length(FOIniMejorada)
[pobla(t,:),fo(t)]=calcularTSP(RutaIniMejorada(t,:),...
    CamionesIniMejorados(t,:),D);
if fo(t)>FOIniMejorada(t)
    pobla(t,:)=RutaIniMejorada(t,:);
    fo(t)=FOIniMejorada(t);
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Poblacion=pobla;
cp=[camiones1;camiones2];
ValFunObj=fo';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% seleccionar dos nuevos padres con probabilidad de mutar
[SelecPadres,ObjPadres]=SelecTorneoVRP(TamPoblacion, Poblacion,ValFunObj);

```

```

Ipadre1=[SelecPadres(1,:)];
Ipadre2=[SelecPadres(2,:)];
% correr el genético de mejoramiento para 4 veces el valor de la generación
%inicial
generaciones=4*generaciones;%ceil(generaciones/2);
end

for gen=1:generaciones

    hijos =CombinacionVRP(Ipadre1, Ipadre2,nciudades);
    hijo1=hijos(1,:);
    hijo2=hijos(2,:);

    % mutacion con el 5%
    if generaciones/tasamutacion>1
        if gen ==ceil((generaciones/tasamutacion)*cmh) % redondeo por exceso
                                                    % a un num entero
            hijosmutados=MutacionVRP(hijo1,hijo2,cmh, nciudades);
            cmh=cmh+1;
            hijo1=[hijosmutados(1,:)];
            hijo2=[hijosmutados(2,:)];
        end
    end

    % asignar camiones a cada hijo
    for k=1:nciudades
        PQh(1,k) = Q(hijo1(k));
        PQh(2,k) = Q(hijo2(k));
    end
    CamPorHijo1(1,:)=FunCamVRP(PQh(1,:),hijo1,nciudades,Cap,ka,cc,sumador);
    CamPorHijo2(1,:)=FunCamVRP(PQh(2,:),hijo2,nciudades,Cap,ka,cc,sumador);
    cph=[CamPorHijo1;CamPorHijo2];
    camionesh1=[];
    camionesh2=[];
    for i=1:nciudades
        if CamPorHijo1(i)==0
            break;
        else
            camionesh1(i)=CamPorHijo1(i);
        end
    end
    for i=1:nciudades
        if CamPorHijo2(i)==0
            break;
        else
            camionesh2(i)=CamPorHijo2(i);
        end
    end
    % hacer 2 opt para mejorar

    tamrutas=size(camionesh1);
    con=1;
    Facum1=0;
    RUTamejorada1=zeros(1,nciudades);
    ite=50;
    for i=1:tamrutas(2)
        nciudadesopt=camionesh1(i);
        rutaopt=hijo1(con:con+camionesh1(i)-1);

```

```

[rutaoptmejorada, fobj]=OptVRPmejoramiento(camionesh1(i), nciudadesopt, D, rutaopt, it
e);
Facum1=Facum1+fobj;
RUTAmejorada1(con:con+camionesh1(i)-1)=rutaoptmejorada;
con=con+nciudadesopt;
end

%%%hijo 2

tamrutas=size(camionesh2);
con=1;
Facum2=0;
RUTAmejorada2=zeros(1, nciudades);
for i=1:tamrutas(2)
nciudadesopt=camionesh2(i);
rutaopt=hijo2(con:con+camionesh2(i)-1);
[rutaoptmejorada, fobj]=OptVRPmejoramiento(camionesh2(i), nciudadesopt, D, rutaopt, it
e);
Facum2=Facum2+fobj;
RUTAmejorada2(con:con+camionesh2(i)-1)=rutaoptmejorada;
con=con+nciudadesopt;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hijo1=RUTAmejorada1;
hijo2=RUTAmejorada2;
% calcular el valor objetivo de cada hijo
ValFunObjhijo1=FunObjVRP(D, cph(1,:), hijo1, nciudades);
ValFunObjhijo2=FunObjVRP(D, cph(2,:), hijo2, nciudades);
fobh=[ValFunObjhijo1; ValFunObjhijo2];

% reemplazar el peor padre por el mejor hijo
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if fobh(1)<= fobh(2) % hijo 1 mejor que hijo2
if ObjPadres(1,1)<=ObjPadres(2,1)
% padre1 mejor que padre2
if (fobh(1)<=ObjPadres(1,1) | fobh(1)<=ObjPadres(2,1))
% hijo1 mejor que padre 1 y mejor que padre2
Poblacion(ObjPadres(2,2),:)=hijo1(1,:); % reemplaza padre 2 por hijo1
cp(ObjPadres(2,2),:)=cph(1,:);
PQ(ObjPadres(2,2),:)=PQh(1,:); % ObjPadres(2,2)=Pos2
ValFunObj(ObjPadres(2,2))=fobh(1);
end
else % padre 2 mejor que padre 1

if (fobh(1)<=ObjPadres(2,1) | fobh(1)<=ObjPadres(1,1))
% hijo1 mejor que padre 2 y mejor que padre1
Poblacion(ObjPadres(1,2),:)=hijo1(1,:);
% reemplaza padre 1 por hijo1
cp(ObjPadres(1,2),:)=cph(1,:);
PQ(ObjPadres(1,2),:)=PQh(1,:);
% ObjPadres(1,2)=Pos1
ValFunObj(ObjPadres(1,2))=fobh(1);
end
end
else
% hijo 2 mejor que hijo1
if ObjPadres(1,1)<=ObjPadres(2,1) % padre1 mejor que padre2

```

```

        if (fobh(2)<=ObjPadres(1,1) | fobh(2)<=ObjPadres(2,1))% hijo2 mejor que
padre 1 y mejor que padre2
            Poblacion(Pos2,:)=hijo2(1,:);
            % reemplaza padre 2 por hijo2
            cp(ObjPadres(2,2,:),)=cph(2,:);
%pos2
            PQ(ObjPadres(2,2,:),)=PQh(2,:);
            ValFunObj(ObjPadres(2,2))=fobh(2);
        end
        else % padre 2 mejor que padre 1
            if (fobh(2)<=ObjPadres(2,1) | fobh(2)<=ObjPadres(1,1))
%hijo1 mejor que padre 2 y mejor que padre1
                Poblacion(ObjPadres(1,2,:),)=hijo2(1,:);
                % reemplaza padre 1 por hijo2
                cp(ObjPadres(1,2,:),)=cph(2,:);% Pos1
                PQ(ObjPadres(1,2,:),)=PQh(2,:);
                ValFunObj(ObjPadres(1,2))=fobh(2);
            end
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% seleccionar dos nuevos padres con probabilidad de mutar
[SelecPadres,ObjPadres]=SelecTorneoVRP(TamPoblacion, Poblacion,ValFunObj);
Ipadre1=[SelecPadres(1,:)];
Ipadre2=[SelecPadres(2,:)];
% hacerlo durante varias iteraciones
% mutación de algunos hijos

end
geninicial=geninicial+1;
end % while
% soluciones con AG
FOIniMejorada
ValFunObj
cp
Poblacion
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%VNS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
% VNS

% ej 1 2 3 4 5

% 2 1 3 4 5
% 3 2 1 4 5
% 4 2 3 1 5
% 5 2 3 4 1

% 1 3 2 4 5
% 1 4 3 2 5
% 1 5 3 4 2

% 1 2 4 3 5
% 1 2 5 4 3

% 1 2 3 5 4

[minimafobj posminfobj]=min(ValFunObj);
incumbenteVNS=minimafobj;

```

```

camionesincumbVNS=cp(posminfobj,:);
inicialincumbenteVNS=Poblacion(posminfobj,:);
RutaVNS=inicialincumbenteVNS;

% nciudades=5;
%
% %inicial=randperm(nciudades);
% inicial=[1 2 3 4 5]

%
contadorVNS=nciudades;
cont2VNS=1;
inicial=Poblacion(posminfobj,:);
inicialaux=inicial;
while contadorVNS>1
    flag=inicial(cont2VNS);
    for i=cont2VNS:(nciudades-1)
        inicial(cont2VNS)=inicial(i+1);
        inicial(i+1)=flag;

        % se evalúa función objetivo
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%55

PQVNS=[];
for i=1:nciudades % demanda de la ciudad correspondiente
    PQVNS(i) = Q(inicial(i));
end
sumador=0;
ka=1;
cc=1;
CamionesVNS=FunCamVRP(PQVNS,inicial,nciudades,Cap,ka,cc,sumador);
FunObjVNS=FunObjVRP(D,CamionesVNS,inicial,nciudades);

if FunObjVNS < incumbenteVNS
    incumbenteVNS=FunObjVNS;
    camionesincumbVNS=CamionesVNS;
    RutaVNS=inicial;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%55

    inicial=inicialaux;
end
contadorVNS=contadorVNS-1;
cont2VNS=cont2VNS+1;
end

%%solucion VNS

RutaVNS
camionesincumbVNS
incumbenteVNS

%%% mejoramiento usando VNS y TSP para cada ruta

```

```

[RUTA,camiones,foacum,camionesconceros]=TSPruta(D,nciudades,RutaVNS,camionesincum
bVNS);
% solucion mejorada
RUTA
camiones
foacum

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%55

tamrutas=size(camiones);
con=1;
Facum=0;
RUTAmejorada=zeros(1,nciudades);
ite=1000;
for i=1:tamrutas(2)
nciudadesopt=camiones(i);
rutaopt=RUTA(con:con+camiones(i)-1);
[rutaoptmejorada,fobj]=OptVRPmejoramiento(camiones(i),nciudadesopt,D,rutaopt,ite)
;
Facum=Facum+fobj;
RUTAmejorada(con:con+camiones(i)-1)=rutaoptmejorada;
con=con+nciudadesopt;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figuraRUTAS=figure
dibujarVRP(RUTAmejorada,Dis,nciudades,camionesconceros)
error=(Facum-MejorSolucion)*100/MejorSolucion

%%%% capacidad de cada camión

for k=1:nciudades
demrutaopt(k) = Q(RUTAmejorada(k));
end
a=0;
for k=1:tamrutas(2)
demandacamiones(k)=sum(demrutaopt((a+1):(a+camiones(k))));
a=camiones(k);
end

tiempo=toc;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Función para guardar un registro de las rutas y la gráfica de la respuesta en
una carpeta de RESULTADOS.
%Obtener el Path de la carpeta Actual
[stat,struc] = fileattrib;
PathCurrent = struc.Name;

%Obtener el Path de la carpeta Actual
[stat,struc] = fileattrib;
PathCurrent = struc.Name;

for n=1:1 % número de carpetas que se desee
%crear las rutas (Path) para carpetas y archivos
FolderName = ['augerat32' num2str(n)];
PathFolder = [PathCurrent '/RESULTADOS/' FolderName];
Nameexcel = [PathFolder '/hoja.xls'];
set(figuraRUTAS,'Name',[FolderName '_sin'],'NumberTitle','off');

```

```

grid on;
title('RUTAS AUGERAT32')
% crear las carpetas para guardar los resultados
mkdir([PathCurrent '/RESULTADOS'], FolderName);
% guardar figura en la carpeta
saveas(figuraRUTAS, [PathFolder '/RUTAS32.fig'], 'fig');

%Guardar datos en archivo EXCEL
titulo = {'MejorSolucion' 'error' 'foacum' 'tiempo'};
datos = vertcat(MejorSolucion,error,foacum,tiempo)';
xlswrite(Nameexcel, titulo, 1, 'A1');
xlswrite(Nameexcel, datos, 1, 'A2');
    titulo = {'camiones'};
    datos = vertcat(camiones)';
xlswrite(Nameexcel, titulo, 1, 'E1');
xlswrite(Nameexcel, datos, 1, 'E2');
    titulo = {'RUTA'};
    datos = vertcat(RUTA)';
xlswrite(Nameexcel, titulo, 1, 'F1');
xlswrite(Nameexcel, datos, 1, 'F2');
end

```

Los resultados de correr el código anterior, generan principalmente una representación grafica de las rutas y la secuencia de clientes correspondientes a cada vehículo. Como ejemplo se indica un resultado para la instancia de 80 clientes:

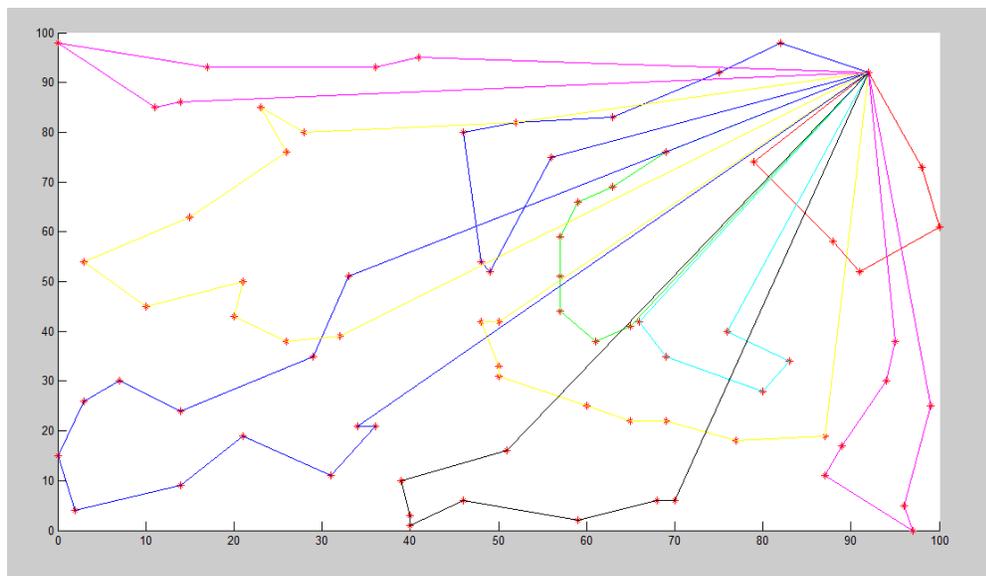


Figura 9-1. Rutas solución VRP instancia de Augerat para 80 ciudades.

```

error =

    5.0227

Elapsed time is 246.604137 seconds.
>> RUTAmejorada

RUTAmejorada =

Columns 1 through 16

    33     5    23    46    51    77    73    71    52    78    30    75     4    59    39    63

Columns 17 through 32

    64    12    24     6    13    45    25     7    31    60    28    32    18    40    65    34

Columns 33 through 48

    16    42    26    76    21    47    61    55    10    56    57    70    66    36    27    48

Columns 49 through 64

    20    58    79    62    17    44    69     9    38     3    35    53    11    72    29    80

Columns 65 through 79

    19    49    15    14     2     8    22    41    43    54    67    68    37    74    50

>> camiones

camiones =

     8     7     6     8    10    12     9     7     5     7

```

Figura 9-2 Secuencia de clientes en solución VRP para instancia de 80 ciudades.

La figura 9-1 indica las rutas solución del problema y en la figura 9-2 se observa el texto generada por el software matlab, en el que hay un error del 6% comparado con el valor óptimo, publicado en las instancias de referencia de Augerat en. [54]. Además de la secuencia de ciudades a visitar y las ciudades a visitar por cada vehículo. Por ejemplo en el vector *camiones* se tiene el primer valor en 8, significando que el primer vehículo visita a los 8 primeros clientes del vector presentado como *RUTAmejorada*, es decir a los clientes 33-5-23-46-51-77-73-71. El segundo vehículo visitaría a los siguientes 7 clientes, etc. Además de que cada ruta debe partir y llegar al depósito, nombrado como cliente 1, el cual no aparece en el vector de *RUTAmejorada*.

9.2 ANÁLISIS DE ARTICULOS RELACIONADOS

The real-time time-dependent vehicle routing problem [8], [57]

RESUMEN:

En el presente artículo, se toma un problema de enrutamiento de vehículos dependientes del tiempo y con ventanas de tiempo, su solución está dada por programación entera mixta al modelo matemático que se describe en tiempo real, así como las rutas y horarios los cuales se toman como variables de decisión, además se tiene en cuenta un retraso en los nodos de salida. Este problema se prueba con 56 puntos, encontrando su valor óptimo.

TÉCNICA DE SOLUCIÓN

Solución del modelo matemático por medio de programación entera y heurísticas de inserción.

CODIFICACIÓN DEL PROBLEMA

La solución del problema según los autores se hace con base en el algoritmo de inserción de clientes.

CALIBRACIÓN DE PARÁMETROS

El desarrollo del modelo matemático consiste básicamente en la adición de restricciones y penalizaciones al modelo de ruteo de vehículos. La solución de este se basa en verificar el costo de introducción un cliente a una ruta verificando su dependencia de tiempo basado en la probabilidad de congestión vehicular, la ventana de tiempo y la demanda del cliente.

RESULTADOS

Este problema se resuelve usando las 56 instancias de máximo 36 clientes según lo reporta Solomon, obteniendo buenos resultados en tiempos prudentes.

En este caso se toma varias instancias para hacer las pruebas correspondientes con el VRP, para posteriormente adicionarle las restricciones de tiempo con la misma metodología del VRP obteniendo resultados que están enmarcados en situaciones reales con valores de tiempo en un problema que se aproxima a la realidad, además de la posibilidad de cambiar las rutas de los vehículos de acuerdo a las restricciones en tiempo real de tráfico y tiempos de ventana de los clientes.

CONCLUSIONES

El RT-TDVRPTW fue estudiado y formulado como un modelo de programación entera mixta, que representa restricciones de tiempo de viaje, demanda y tiempos de espera o de ventana en tiempo real generando un algoritmo que actualiza las rutas en tiempo real con pequeñas modificaciones al problema básico de VRP.

En el modelo que se plantea, es importante resaltar que la hora de salida de los vehículos es tomada como una variable de decisión, la cual permite penalizar y optimizar el uso de la ruta en ciertos horarios.

Además de las restricciones en el modelo se plantea un nodo crítico penalizándolo para no optar por la introducción de este cliente a la ruta. este concepto de nodos críticos distingue la ventana de tiempo en alto grado en tiempo real con el problema tradicional.

ANÁLISIS

El artículo permite visualizar el problema de ruteo de vehículos con dependencia de tiempo en situaciones cercanas a la realidad, por lo que es de gran importancia en el desarrollo del proyecto, además permite caer en cuenta de la dificultad tan grande en la solución del modelo matemático planteado como programación entera. Así se nota como las soluciones son muy buenas en la medida de las posibilidades computacionales, sin embargo para instancias mayores sería imposible dar solución a este problema por métodos exactos, así se pretende implementar este problema como una codificación metaheurística que resuelva el problema con un número mucho mayor de instancias y con valores cercanos al óptimo en tiempo computacional pertinente.

Para el desarrollo del proyecto se pretende usar la codificación algorítmica de las restricciones cambiantes en tiempo real de los valores de tiempo que penalizan las rutas según el horario.

Adicionalmente en este artículo se muestra cómo se puede aplicar el problema de ruteo de vehículos en casos reales, como se pretende con este tipo de problemas que en el caso se aplicaría al modelo de visitas para entrega de productos lácteos en la ciudad de San Juan de Pasto.

Time dependent vehicle routing problem with a multi ant colony system. [6]

RESUMEN:

En el artículo relacionado, se resuelve el problema TDVRP en el que el tiempo empleado para atravesar cada arista, depende de la hora del día en que se realiza el viaje, además se consideran flotas de vehículos con capacidad fija. El método de optimización consiste en la búsqueda de soluciones que minimicen dos objetivos jerárquicos: el número de visitas y el tiempo total de viaje.

OBJETIVO

Notar como la restricción de tiempo en el problema de ruteo de vehículos hace que la solución factible en el VRP sea infactible en el VRPTD.

TÉCNICA DE SOLUCIÓN

Para la solución del problema se implementó un Algoritmo de colonia de hormigas y ruta mínima entre dos pares de puntos.

CODIFICACIÓN DEL PROBLEMA

El problema se codifica con vectores aleatorios de rutas o recorridos por hormigas artificiales que buscan el óptimo, estas hormigas tienen una función de eliminación y creación de la feromona en función del tiempo, la cual permite obtener una ruta por cada hormiga de forma cercana al óptimo.

POBLACIÓN INICIAL

La población de rutas iniciales para las hormigas se hace de manera aleatoria. Luego de encontrar rutas con las hormigas artificiales se realiza una corrida del algoritmo para mejorar la solución factible usando algoritmos de búsqueda local.

FUNCIÓN DE ADAPTACIÓN:

En este caso la función de adaptación del algoritmo está dada de acuerdo a la probabilidad con que se evapora o conserva la feromona, la cual está en función del horario en que se dan las rutas.

CALIBRACIÓN DE PARÁMETROS

Para esto se usa una ecuación de probabilidad con que se actualiza la feromona en cada cliente en función de constantes de peso y el tiempo.

RESULTADOS

El algoritmo tiende a favorecer a las rutas con clientes más alejados ya que posiblemente viajaban en períodos de tiempo con velocidades más altas, y los clientes con distancia más corta en períodos de tiempo con velocidades más bajas. Las mejores soluciones conocidas para el modelo no dependiente de tiempo se utilizan en el contexto del modelo dependiente del tiempo, su factibilidad y optimalidad podrían cambiar drásticamente en función de los horarios.

CONCLUSIONES

Se presenta un modelo dependiente del tiempo para el vehículo de enrutamiento basado en colonia de hormigas con un componente dependiente del tiempo para la actualización de

feromonas. El algoritmo también está soportado por los procedimientos de búsqueda local, y se mostró la forma en que es posible mantener constante en el tiempo la búsqueda de movimientos adecuados para el recorrido de las hormigas.

Por otra parte se demuestra que en la mayoría de los casos con la mejor solución conocida en el VRP, no se puede afirmar que esta sea adecuada en el VRP con dependencia de tiempo.

Para aplicaciones reales se debe tener muy en cuenta los horarios en que se hacen los recorridos para la búsqueda de valores óptimos en el ruteo de una situación real.

ANÁLISIS

En este artículo se nota como una metaheurística encuentra una buena solución a un problema combinatorial difícil, y como el tiempo influye notablemente en la solución de un problema de ruteamiento de vehículos.

De este modo para el desarrollo del proyecto se hace uso de la información del algoritmo de hormigas en la generación de soluciones al VRP iniciales para luego generar una búsqueda intensiva con otros métodos heurísticos y Metaheurísticos que encuentren mejores soluciones para instancias de un gran número de combinaciones.

Por otra parte en el proyecto a desarrollar se tendrá en cuenta el método de actualización de feromona pero en rutas mucho más grandes con instancias de al menos 80 clientes.

TESIS DOCTORAL Optimización heurística económica aplicada a las redes de transporte del tipo VRPTW. [1]

RESUMEN:

El propósito de la tesis consiste en la presentación de un modelo económico de distribución de mercancías que generalice los problemas de rutas sometidos a restricciones temporales de servicio “vehicle routing problem with time windows” (VRPTW) y de un conjunto de técnicas heurísticas y metaheurísticas capaces de resolverlo eficientemente.

OBJETIVO

Diseñar nuevos algoritmos que permitan resolver problemas de transporte y que sean robustos, flexibles y lo suficientemente rápidos como para abordar casos que, alejándose de los esquemas teóricos, se acerquen lo más posible a las situaciones reales.

TÉCNICA DE SOLUCIÓN.

En esta tesis se analiza un conjunto variado de técnicas de resolución basadas en la búsqueda secuencial por entornos que se aplicarán al problema VRPTW básico. Asimismo se ensayan distintas variantes que surgen al modificar los parámetros que definen cada una de las estrategias. La optimización heurística de un problema combinatorio comienza con una solución, o un conjunto de ellas que, mediante cambios guiados estratégicamente, llevan a un resultado satisfactorio, aunque no necesariamente el óptimo.

CODIFICACIÓN DEL PROBLEMA

En este trabajo se propone una heurística denominada HESECOR (heurística económica y secuencial de construcción de rutas) que permitirá la generación de soluciones tanto para el VRPTW como para las generalizaciones contempladas en el modelo propuesto. Este algoritmo constituye la base del desarrollo de las metaheurísticas empleadas posteriormente en la tesis.

FUNCIÓN DE ADAPTACIÓN: (CRITERIO DE EVALUACIÓN)

Todas las técnicas que se presentan parten de soluciones iniciales generadas con la heurística HESECOR. Se han propuesto diez estrategias de búsqueda basadas en distintas familias de metaheurísticas, y otras cuatro híbridas de las anteriores.

CALIBRACIÓN DE PARÁMETROS

Todos los parámetros se basan en la comparación de algoritmos clásicos con metaheurísticas y la formación de híbridos entre metaheurísticas, las cuales se basan en generar soluciones iniciales y posteriormente usar algoritmos como Búsqueda en vecindario variable, búsqueda Tabú y recocido simulado para mejorar las soluciones de acuerdo al algoritmo de economías planteado en la tesis, las soluciones obtenidas con diferentes métodos cuando no mejoran se usan algoritmos para empeorar algunas rutas y continuar iterando hasta encontrar mejoras o valores cercanos al óptimo.

RESULTADOS

El algoritmo de las economías propuesto en la tesis mejora las soluciones generadas por algoritmos clásicos en tiempo y en valor en la función objetivo y sobre todo la opción de solución a problemas genéricos básicos que se desprenden del VRPTW.

El autor toma como referencia la instancia de Solomon y realiza comparaciones con los diferentes métodos heurísticos y metaheurísticos obteniendo resultados como se indican en los anexos del documento.[1]

ANÁLISIS

La tesis de Víctor Yepes, menciona el uso de una metaheurística basada principalmente en los ahorros en la selección de rutas, esto lo logra básicamente retirando algunos puntos o nodos y resolviendo un problema de ruteo con los diferentes algoritmos existentes, posteriormente usa un método de inserción de los nodos retirados y se mide la función objetivo; para la inserción de los nodos se usa algoritmos como recocido simulado principalmente y se hace una búsqueda en los entornos con el fin de salir de óptimos locales.

Los resultados que obtiene son de buena calidad según los que se tiene como referencia para este caso, además de considerar muchos aspectos cercanos a la realidad como son el uso de flotas heterogéneas con capacidad no fija en los vehículos y las ventanas de tiempo variables, además de considerar a la función objetivo como la rentabilidad económica que tendría el uso de las rutas dependientes de distancias, satisfacción del cliente en la medida en que se entregue los productos a tiempo por medio de factores como velocidad de carga, jornadas laborales, uso múltiple de vehículos, asimetría en la duración de los viajes, congestión vehicular, horas extras y ruptura en la llegada al depósito. Todos estos elementos que se asocian con las restricciones del modelo matemático generan gran dificultad en la solución del problema, por lo que en la tesis se soluciona usando una penalización a la función objetivo que reduce o aumenta su rentabilidad económica.

Para el caso del proyecto a realizar se toma como referencia algunas de las técnicas heurísticas y metaheurísticas con mayor relevancia en la generación de una población inicial del algoritmo genético para el problema de ruteo de vehículos dependientes del tiempo y optando por la adición de la penalización de la función objetivo o aumentando un factor a la matriz de distancias que permita generar diferentes costos en la medida en que los horarios en los transportes cambien con valores de probabilidad según la congestión vehicular.

Por otra parte en el proyecto a realizar se generara la solución a un problema tomado totalmente de la realidad para una empresa de distribución de productos lácteos en la ciudad de San Juan de Pasto, para lo cual se establecen los puntos geo referenciados y se establece el modelo usando una matriz de costos por distancia y por penalización de esta según el horario en el que se genera la ruta.

Modelado matemático para el análisis y mejoramiento de los métodos aproximados para la solución de los problemas de enrutamiento, asignación de horarios y distribución de cargas: una aplicación a la entrega de productos embotellados. [14]

RESUMEN:

Este trabajo presenta un modelo de programación entera para el problema RoSLoP (Routing-Scheduling-Loading Problem) en una compañía caso de estudio. La meta de RoSLoP es obtener una solución para el proceso de entrega de productos embotellados, minimizando el número de vehículos y la distancia de recorrido de los vehículos para satisfacer los pedidos de los clientes. RoSLoP se ha formulado a partir de dos problemas clásicos de optimización combinatoria: el problema de enrutado de vehículos (Vehicle Routing Problem o VRP) y el problema de empaqueo de objetos (BPP).

OBJETIVO

1. Formular un nuevo modelo matemático que incorpore las variantes del problema VRP involucradas en el problema RoSLoP de la empresa caso de estudio.
2. Elaborar un estudio comparativo de los métodos de solución aproximados THSA y SBH-ARHC desarrollados para el problema RoSLoP.

TÉCNICAS DE SOLUCIÓN

El modelo matemático desarrollado para RoSLoP, se formuló utilizando las restricciones de los modelos creados en la literatura para la construcción de soluciones a las variantes de VRP. Lo cual involucra la solución de un problema de optimización con satisfacción de restricciones (CSP) en el caso de un método exacto, así como la aplicación de formulismos que permiten la reducción de la dimensión del problema y la validación de las restricciones cuando la alternativa de solución es aproximada, como ocurre en la implantación de los métodos aproximados THSA y SBH-ARHC en el sistema de administración logística de la empresa caso de estudio.

CODIFICACIÓN DEL PROBLEMA

Para dar solución al problema del VRP, el autor toma el modelo matemático clásico del VRP y como principal objetivo menciona el uso mínimo de vehículos para cubrir las rutas con la menor distancia y el menor tiempo de viaje tomando la base de datos de la empresa con la información de distancias y tiempos además de la demanda de cada uno de los clientes, posteriormente se toma el modelo matemático planteado en la tesis junto con una instancia del problema para realizar una comparación usando un software especializado como es LINDO API 4.0.

RESULTADOS

Las métricas aplicadas para los problemas de solución de problemas VRP son: número de vehículos utilizados y distancia recorrida, parámetros de interés en la empresa del caso de estudio. Por ello se desea tener una solución de referencia para medir el desempeño de los métodos aproximados de la variante rich VRP asociada a RoSLoP.

CONCLUSIONES

Se creó un método de solución exacto para la solución de RoSLoP, el cual se fundamenta en dos problemas de optimización clásicos: ruteo de vehículos VRP y empaçado de productos BPP. Siendo objeto de estudio de este trabajo la solución de una variante rich VRP como disminución de costo por número mínimo de vehículos a usar en las rutas asociadas al caso de estudio. Se obtuvo, del análisis comparativo que los métodos heurísticos obtienen soluciones de 100% y 98% de aproximación en los parámetros de VRP: vehículos utilizados y distancia recorrida respectivamente, respecto a la solución óptima obtenida como referencia por el método exacto de este trabajo.

ANÁLISIS

En este proyecto se nota claramente como el modelado matemático de los problemas combinatoriales permiten resolver de manera exacta o casi exacta cada restricción, sin embargo el número de ciudades a visitar sigue siendo limitado por la generación de las múltiples combinaciones y lo imposible de resolver por un procesador.

Por otro lado es importante recalcar como un modelo matemático permite generalizar los diferentes tipos de problema con la acción de adicionar o eliminar líneas de ecuaciones en el modelo ya sea como restricciones o coeficientes en la función objetivo, de este modo se puede acercar con mayor precisión a la realidad de la distribución de productos de las empresas, además de permitir agregar diferentes factores que puedan influir en el tiempo a los casos particulares de las empresas como son el tiempo de ventana, satisfacción del cliente, demandas heterogéneas, capacidades de los vehículos ya sean fijas o no, entre otros elementos.

Para el caso del proyecto a realizar se usa algunos formatos del modelado matemático para generar restricciones en las rutas como es el tiempo de recorrido de las rutas que se involucra junto con la distancia en el costo de la ruta, así se pretende usar un factor de estandarización entre tiempo y distancia para generar un costo que dependa tanto de la distancia de recorrido como del horario en el que el vehículo visita a los clientes como se desarrolla en la tesis de José Delgado.[14].

También es importante recalcar que el desarrollo de software permite no solo adicionar restricciones particulares en una empresa, sino que presenta flexibilidad en el manejo de dichas restricciones, así se puede generalizar el problema y su solución o particularizar de acuerdo a los requerimientos de la empresa, midiendo el tiempo de ejecución del programa y la manipulación de diferentes variables que no las contempla en un software comercial.

Vehicle routing under time-dependent travel times: The impact of congestion avoidance. [7]:

La congestión del tráfico diario constituye un problema importante para las empresas, tales como los proveedores de servicios logísticos y las empresas de distribución. La causa de llegadas tarde de clientes genera costos adicionales para la contratación de los conductores de camiones. Tales costos causados por la congestión del tráfico pueden reducirse teniendo en cuenta que se puede evitar la congestión del tráfico la cual es predecible antes de hacer el viaje. En la literatura, se proponen diversas estrategias para evitar la congestión del tráfico, tales como la selección de rutas alternativas, el cambio de las secuencias de visita de los clientes, y el cambio de las asignaciones de vehículo a los clientes.

En este artículo se presenta el desarrollo de una serie de rutas para vehículos con instancias de problemas en redes reales por carretera, y un modelo de velocidad que refleja los elementos clave de la congestión del tráfico de la hora pico. Los casos se resuelven para diferentes niveles de congestión utilizando el algoritmo de Dijkstra modificado y una heurística de programación dinámica restringida.

OBJETIVO

Encontrar la solución del problema de ruteo de vehículos con dependencia del tiempo estableciendo un modelo de velocidad de tráfico para reducir los costos de las rutas en distancia y tiempo.

TÉCNICAS DE SOLUCIÓN

Los autores proponen el método de solución por medio del algoritmo modificado de Dijkstra, el que permite encontrar soluciones con el vecino más cercano, esto con el fin de resolver un TSP para cada ruta de forma rápida para que no afecte el cambio inmediato en el tráfico vehicular, así se selecciona una ruta para un tiempo definido y se cambia según el horario junto con la velocidad de flujo de los vehículos y del tráfico.

CODIFICACIÓN DEL PROBLEMA

La propuesta de los autores consiste en modelar la velocidad con que se desplaza el vehículo sobre la ruta en función del tiempo incorporando la hora pico y la congestión de tráfico.

CRITERIO DE EVALUACIÓN:

Para cada situación se resuelve un modelo de velocidad dependiente del tiempo denominado TDSPP y luego un problema de ruteo dependiente del tiempo VRPTD usando métodos de solución clásicos como el algoritmo de Dijkstra y la ruta más corta con el fin de resolver de forma general el VRP y con cambios simples solucionar la dependencia de tiempo, además se considera el tiempo de respuesta en la medida que este no debe afectar al cambio de horario mientras se resuelve el

problema, posteriormente se hace un seguimiento de los horarios de salida en cada nodo alcanzado con el fin de determinar los tiempos de viaje en función del tiempo u horario y asignar etiquetas de los nodos que necesiten ser actualizados.

RESULTADOS Y CONCLUSIONES:

Los autores proponen cuatro escenarios diferentes en los que las congestiones de tráfico se evitan con un promedio creciente. Para esto se ha propuesto un modelo que considera la velocidad en redes de tráfico reales por carretera, durante las horas pico que causan mayor congestión de tráfico. Este modelo de velocidad proporcionó una plataforma para probar el impacto de los diferentes escenarios de la realidad. Los resultados indicaron gran confiabilidad en la ruta al presentarse congestiones de tráfico en ciertas horas. Sin embargo, si se modela el problema del VRP con tiempos de viaje independientes del horario, la confiabilidad permite disminuir el costo visto en la disminución del número de rutas de vehículos y el tiempo de actividad total, además cuando se considera el problema de ruteo dependiente del tiempo en los niveles más altos de evitación de la congestión, las medidas de costo se pueden reducir significativamente puesto que se eliminan los vehículos que permanecen en la congestión, reduciendo los costos totales.

ANÁLISIS

Los autores mencionan que debido a la creciente cantidad de tráfico y una limitada capacidad de la red de carreteras, la congestión de tráfico se ha convertido en un fenómeno cotidiano causando costos elevados en la contratación de conductores de vehículos pesados en las empresas de logística y distribución.

La congestión del tráfico puede tener varias causas. Algunas muy predecibles, como la gran cantidad de tráfico de pasajeros durante las horas máximas diarias y otras son menos previsibles, tales como el clima o los accidentes de tráfico.

Dado que las congestiones en hora pico son muy predecibles y que constituyen un 70 a 87% de todos los retrasos los autores se centran en evitar congestiones de tráfico en la hora pico evitando estar en el lugar equivocado en el momento equivocado. Las estrategias para lograrlo son cambiar la secuencia de la visita de un vehículo, quitar a un cliente de una ruta de vehículos y se inserta en un complemento de otra. Estas estrategias de evitación de la congestión se pueden optimizar mediante la resolución de un VRP con tiempos de viaje dependientes del tiempo (TDVRP). Otra estrategia para evitar congestiones de tráfico es seleccionar una ruta alternativa entre dos clientes en horas pico. Esta estrategia implica que la ruta entre dos clientes depende de la hora de salida elegida, que puede optimizarse mediante la resolución de un problema del camino más corto con los tiempos de viaje en función del tiempo (TDSPP).

Para el proyecto se pretende tener en cuenta las estrategias de solución del problema de VRP para luego adicionar las restricciones del tiempo pero usando instancias de prueba de 80 a 100 clientes, para posteriormente poner en prueba en un caso real para la distribución de productos lácteos en la ciudad de San Juan de Pasto. Además la solución inicial será planteada por diferentes métodos heurísticos para potencializar las soluciones por medio de algoritmos genéticos.

Time Dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms. [5]

RESUMEN:

El problema TDVRP se define como una flota de vehículos con capacidad fija que presta servicios a clientes con demandas fijas, el vehículo parte del depósito y visita los diferentes clientes de tal manera que el tiempo total de las rutas se reduce al mínimo. El tiempo de viaje entre dos clientes o entre un cliente y el depósito depende de la distancia entre los puntos y la hora del día.

Las características y propiedades de este problema semejantes a la del VRP, o incluso al problema del agente viajero permiten el uso de algoritmos clásicos para su solución con algunas modificaciones que generalmente se basan en la heurística del vecino más cercano.

OBJETIVO

Minimizar el costo de las rutas de acuerdo a la distancia y la hora del día.

TÉCNICAS DE SOLUCIÓN

La solución planteada por los autores se remite al modelado matemático del problema de TSP con dependencia de tiempo y luego el modelo de VRP y la consideración de la restricción de la dependencia del horario en que se genera la ruta. Para esto los autores generan una probabilidad de los horarios con mayor dificultad de tráfico y lo asignan al modelo, para realizar su solución por medio de una heurística planteada como una secuencia de pasos que resuelven el modelo del TSP y lo generalizan para un VRP. Finalmente la heurística que plantean se basa directamente en los métodos de programación lineal entera, que para este caso usan el método de los planos de corte.

CODIFICACIÓN DEL PROBLEMA

El modelo matemático contempla una matriz de costos que depende de la distancia en que se encuentren los clientes y en qué tiempo se puede llegar a estos, es decir la velocidad a la cual puede acceder un vehículo que parte desde el depósito continua por la ruta y retorna nuevamente al depósito.

El problema se resuelve por programación entera

FUNCIÓN DE ADAPTACIÓN:

La función de medida del problema consiste básicamente en la obtención del valor mínimo de tiempo que tarda cada vehículo en recorrer su ruta como un problema de TSP.

CALIBRACIÓN DE PARÁMETROS

Para este modelo los autores plantean un valor de tiempo entre viaje desde cada cliente y el depósito y de acuerdo al horario, con eso se genera la matriz de costos, para lo cual generan soluciones de hasta máximo 25 clientes comparando los valores óptimos según las instancias de VRP propuestas por Christofides.

RESULTADOS

Los autores realizan varias corridas de los algoritmos para solucionar el modelo matemático usando técnicas de planos de corte encontrando soluciones cercanas al óptimo para problemas de TSP y VRP en tiempos adecuados para instancias de 10, 15, 20 y 25 clientes.

CONCLUSIONES

Según los autores, la principal conclusión radica en el tiempo de respuesta del algoritmo de planos de corte de la programación entera. En el caso de instancias pequeñas el algoritmo encuentra buenas soluciones pero para aplicarse a instancias con un número mayor a 25 clientes genera tiempos de cómputo muy altos.

Los autores encuentran que existe gran dificultad en la restricción temporal y no es posible generalizar un modelo de TSP o VRP simples que soluciones el problema de rutas con dependencia del horario, puesto que el tiempo es una restricción dinámica y se debe considerar la solución del problema real en el momento en que se genere el recorrido del vehículo, es decir que se debe correr el algoritmo según la hora en que se requiera.

La consideración del tiempo en las rutas mejora en la realidad los costos de viaje de forma notable, aunque en algunos casos como accidentes de tránsito y congestión por factores climáticos no es fácil de considerar en el modelo.

ANÁLISIS

Para el desarrollo del proyecto los autores generan el interés por el uso de la restricción de tiempo como la conformación de una matriz de costos que no sea únicamente en función de la distancia entre los clientes, sino de la velocidad con que se accede a ellos, para esto se apoya en el concepto de función objetivo propuesta, sin embargo se nota fácilmente como el problema combinatorial no permite realizar un algoritmo de programación entera para un número grande de instancias,

para lo cual en el proyecto se desarrollara métodos metaheurísticos que permitan considerar un número de clientes de 80 a 100 o más.

Aplicación Heuristic, Exact and Hybrid Approaches for Vehicle Routing Problems [17]

RESUMEN:

El C137oblema137RP es un problema de optimización combinatoria clásica que se propuso a finales de 1950 y sigue siendo uno de los más estudiados en el campo de la investigación de operaciones. El gran interés en el VRP es debido a su importancia práctica, así como la dificultad en la solución de ella. Este trabajo aborda heurísticas exactas e híbridos para la solución de las diferentes variantes de la VRP, a saber: VRP capacitados (CVRP), Open VRP (OVRP), asimétrica CVRP (ACVRP), VRP con recogida y entrega simultánea (VRPSPD), VRP con Pickup mixto y Entrega (VRPMPD), Multi-Depot Vehicle Routing Problem, con recogida y entrega Mixto (MDVRPMPD) y flota heterogénea VRP (HFVRP). Una extensa revisión de la literatura se realiza para todas estas variantes, centrándose en los principales aportes de cada obra.

El método de solución exacto es un algoritmo de Ramificación y planos de corte que también está diseñado para resolver el VRPSPD / VRPMPD.

Las pruebas se hacen con los valores de instancias de hasta 200 clientes.

Además los autores proponen un algoritmo heurístico para resolver los VRP, dicho algoritmo lo denominan ILS-RVND, el cual se basa en la búsqueda local iterada (ILS) y hace uso de un algoritmo de descenso variable con zona aleatoria (RVND) en la fase de búsqueda local. Por último, un algoritmo híbrido que incorpora un conjunto de particiones en el enfoque heurístico ILS-RVND para la solución de 8 de los VRP tratados en este trabajo. La heurística desarrolla algoritmos híbridos y se prueban en cientos de casos de referencia y los resultados obtenidos son, en promedio, altamente competitivos.

OBJETIVO

- Revisión de los VRP, que describen algunas aplicaciones prácticas y métodos de soluciones propuestos en la literatura.
- Desarrollar métodos exactos de problemas de ruteo con entrega y recogida.
- Desarrollar un marco general heurística capaz de resolver una gran clase de VRP.
- Desarrollar un algoritmo general híbrido capaz de resolver una gran clase de VRP.

TÉCNICAS DE SOLUCIÓN

El autor en la tesis de doctorado se basa principalmente en la solución por métodos exactos de programación lineal entera al modelo matemático, para ello usa el método simplex para la generación de la solución inicial, a la que posteriormente le realiza un método de ramificación y

planos de corte aplicados a la búsqueda de un valor óptimo con algoritmos de búsqueda local en el entorno de la solución encontrada por el método simplex.

CODIFICACIÓN DEL PROBLEMA

La codificación del problema está basada directamente en el modelo matemático del VRP, con la adición de restricciones como la recogida y entrega de productos, Flota homogénea y heterogénea, demanda y capacidad de los vehículos. Todo en conjunto con el fin de resolver un problema general que contenga varias formas del problema particular como puede ser la flota homogénea o VRP con capacidad fija, entre muchos otros.

CONCLUSIONES

Esta tesis aborda enfoques heurísticos, exactos e híbridos del problema de ruteo de vehículos VRP. Se prueban las soluciones óptimas de 30 problemas abiertos y fueron capaces de demostrar la optimalidad de 7 problemas abiertos, además de encontrar buenas soluciones con problemas de hasta 200 clientes.

ANÁLISIS

La tesis presenta gran contenido de información acerca del modelamiento matemático del problema y como los algoritmos básicos de búsqueda local como 2 OPT, entre otros permiten mejorar la solución inicial sin generar problemas de tiempos computacionales y acercándose al óptimo con algoritmos heurísticos. Por otra parte se tiene en cuenta el uso de algoritmos heurísticos como swap, inserción del más barato, intercambios de nodos y aristas, entre otros para sacar al problema del sesgo sobre un óptimo local y generar la búsqueda en el espacio de soluciones y llegar a otro valor que mejore la función objetivo del problema.

Para el desarrollo del proyecto, esta tesis ayuda a la revisión del estado del arte sobre los problemas y modelamiento matemático en el ruteo de vehículos, sin embargo la metodología de solución que se presenta no está basada directamente en el modelamiento sino en la aplicación de metaheurísticas que permitan obtener soluciones rápidas a problemas complejos con gran cantidad de clientes.

The Time-Dependent Pollution-Routing Problem [18]

RESUMEN:

El problema TDPRP consta del ruteo de una flota de vehículos con el fin de servir a un grupo de clientes y la determinación de las velocidades en cada tramo de las rutas. La función de costo incluye los costos de combustible, las emisiones y el conductor, teniendo en cuenta la congestión

de tráfico que, en las horas pico, restringe significativamente las velocidades del vehículo y aumenta las emisiones.

En el artículo se describe una formulación de programación lineal entera del TDPRP y se proporciona ejemplos ilustrativos para motivar el problema y dar ideas acerca de las ventajas y desventajas que implica. También proporciona una caracterización analítica de las soluciones óptimas para una versión de un solo arco del problema, la identificación de las condiciones bajo las cuales es óptimo.

Con base en los resultados analíticos se presenta un algoritmo de optimización de la velocidad en una ruta fija y el desempeño computacional de instancias de referencia junto con el procedimiento de optimización de la velocidad.

OBJETIVO

Solucionar el problema del ruteo de vehículos dependientes de tiempo considerando la medida de contaminación dada por la velocidad de desplazamiento de la ruta debida a la congestión vehicular.

TÉCNICAS DE SOLUCIÓN

Programación lineal entera.

CODIFICACIÓN DEL PROBLEMA

El problema se modela como un PRP (Pollution routing problem) definido como una extensión del problema clásico de ruteo de vehículos con ventanas de tiempo VRPTW.

De esta forma los autores generan restricciones en el modelo dadas por la velocidad con que el vehículo cumple la ruta, además de hacer correspondencia entre la cantidad de kilómetros de recorrido y la velocidad a la cual los recorre con la cantidad de contaminantes dadas por el consumo de combustible.

FUNCIÓN DE ADAPTACIÓN: (CRITERIO DE EVALUACIÓN)

Los autores proponen como función objetivo la minimización del uso de un factor de consumo de combustible, puesto que este depende de la velocidad de flujo, y las congestiones vehiculares, para ello plantean el factor de consumo dependiente de varios factores como donde la relación de masa de combustible y aire, el valor calorífico de un combustible diésel típico, un factor de conversión de gramos a litros, el factor de fricción del motor, la velocidad del motor, el desplazamiento del motor, la densidad del aire, la superficie frontal, el peso en vacío del vehículo,

la constante gravitacional, el ángulo de la carretera, coeficientes de resistencia aerodinámica y la resistencia a la rodadura, entre otros.

Con este factor se constituye la matriz de costos para evaluar la función objetivo.

RESULTADOS

Se toma distancias reales en ciudades seleccionadas al azar en el Reino Unido con 10, 15 y 20 nodos. Todos los experimentos se llevaron a cabo mediante el uso de CPLEX 12.1 en un servidor con 2,93 GHz y 1,1 Gb RAM. Las ventanas de tiempo y tiempos de servicio, se generan aleatoriamente.

Los autores analizan el problema de ruteo de vehículos con ventanas de tiempo y dependiente del tiempo u horario, el cual considera los vehículos que viajan en dos períodos sucesivos de la congestión y de flujo libre, respectivamente, y de manera explícita las cuentas de consumo de combustible que aumenta considerablemente cuando los vehículos circulan a baja velocidad. Dado que la cantidad de las emisiones de un vehículo es proporcional a la cantidad de combustible consumido, el enfoque de modelización adoptada en este trabajo la solución con rendimientos de menores emisiones de gases de efecto invernadero.

ANÁLISIS

En el desarrollo del proyecto se nota como el modelado matemático genera un gran aporte al comportamiento del problema, sin embargo la técnica de solución exacta no es la mejor en casos en que los clientes se presentan en cantidades mayores a 25, para esto se plantea la solución del problema aproximado usando metaheurísticas, aunque se considera en gran medida el uso de restricciones del tiempo como función de la velocidad con que se desplaza el vehículo y la congestión presente en las vías para formar la función objetivo cercana a la realidad de la distribución de productos en la ciudad de San Juan de Pasto.

The Time-Dependent Vehicle Routing Problem with Soft Time Windows and Stochastic Travel Times. [19]

RESUMEN:

En este trabajo se estudia el problema de ruteo de vehículos dependientes de tiempo, tomando los tiempos de viaje de forma probabilística. En este problema, los clientes tienen ventanas de tiempo un tanto permisibles.

El modelo matemático que describe este problema se basa en la eficiencia de la prestación del servicio y la fiabilidad que se genera en los clientes dependiendo si los tiempos de servicio incluyen o no.

La solución de este problema está dada por dos metaheurísticas, la Búsqueda Tabú y la técnica de Búsqueda en entorno variable, Llevando a cabo experimentos para casos de problemas conocidos y realizando análisis exhaustivos sobre los resultados numéricos en términos de tiempo de cálculo y de calidad de la solución, llegando a la conclusión de que los resultados pueden ser soluciones muy buenas para ser implementadas en el entorno de la vida real.

OBJETIVO

Solucionar el problema de ruteo de vehículos dependientes del tiempo donde el tiempo de viaje en las rutas sea determinado por medio de un proceso estocástico.

TÉCNICAS DE SOLUCIÓN

Modelado matemático, metaheurística de búsqueda Tabú y búsqueda en entorno variable.

CODIFICACIÓN DEL PROBLEMA

Los pasos para encontrar la solución del problema son:

1. Construir una solución factible inicial con respecto a los tiempos probables de viaje dependientes del tiempo.
2. Mejorar la solución generada en el paso 1 mediante el uso de un método de búsqueda tabú TS con respecto al costo total ponderado.
3. Mejorar la solución generada en el paso 1 mediante el uso de un método de búsqueda en entornos variables adaptativos ALNS con respecto al costo total ponderado.
4. Aplicar un método de post-optimización de las soluciones obtenidas en el paso 2 y en el paso 3.

CRITERIO DE EVALUACIÓN:

Los autores toman en consideración el uso de instancias de problemas con 100 clientes y un depósito según lo propone Salomón en el año de 1987. Se toma el valor de la función objetivo como la minimización del costo dado por la relación de la distancia y el tiempo que tarda ir de un cliente a otro, además de tener en cuenta restricciones como las ventanas de tiempo y la capacidad fija de los vehículos.

RESULTADOS

En este trabajo, se considera un VRP con ventanas de tiempo suaves en las que los tiempos de viaje son modelados con respecto a la dependencia del tiempo y la probabilidad de congestión. La

solución inicial construida en la primera fase se ha mejorado en la fase de enrutamiento que es manejado por un TS y un ALNS. En la tercera fase, se aplica un método de post-optimización a las soluciones obtenidas por las metaheurísticas. Formulando la hora de llegada, distribuciones tanto exactamente (sin tiempos de servicio) y aproximadamente (con tiempos de servicio). Llevando a cabo experimentos computacionales en casos de problemas conocidos. Los resultados indican que las formulaciones propuestas en este documento estiman con fiabilidad las distribuciones y los valores esperados empleados en el modelo matemático. A pesar de que este modelo es bastante complejo y las distribuciones del tiempo de llegada son muy complicados, se tiene un procedimiento de solución eficaz que ofrece muy buenas soluciones finales en un período razonable de tiempo.

ANÁLISIS

El uso de metaheurísticas en un problema de ruteo con varios clientes permite generar soluciones de buena calidad en un buen tiempo computacional, como lo indican los autores en el artículo en el que trabajan con instancias de hasta 100 clientes. En el caso del proyecto en particular se toma la consideración de la matriz de costos en función de la velocidad con que se completa la ruta, así se permite tener en cuenta el costo tanto por distancia como por ocurrencia de congestión vehicular dependiente del horario en que se genere la ruta.

9.3 FUNCIONES PARA ASIGNACIÓN DE DEMANDAS A LOS DIFERENTES SECTORES

```

% comparacion y codificaion de sectores para el aporte de cada ruta... en
% la demanda de cada vehiculo al sector
%for i=1:24 % 24 rutas
srl=zeros(1,length(r24));
srr=[];
con=1;
    for j=1:length(r24) % longitud de cada ruta
        for k=1:154 % 154 sectores
            if r24(j,')==sector(k,:)
                srl(j)=k;
                srr(con)=k;
                con=con+1;
            end
        end
    end
end

```

Figura 9-3. Generación de código para el sector en cada punto de la ruta.

La función de la figura 9-3, se encarga de recorrer cada uno de los 154 sectores por cada uno de los 24 vehículos para asignar la pertenecía del vehículo o no en el cubrimiento del sector.

```

sec=zeros(154,3);
for i=2:154
    con=1;
    for filas=1:15
        for col=1:24
            if i==sectorruta(filas,col)
                sec(i,con)=col;
                con=con+1;
            end
        end
    end
end
end

```

Figura 9-4. Función para encontrar las rutas que aportan a cada sector.

El código de la figura 9-4 se encarga de realizar el recorrido de cada vehículo por los determinados sectores.

```

dema=0;
tiendSec=0;
for i=2:154
    for j=1:6 % maximo numero de sectoes por ruta
        if sec(i,j)~=0
            dema=dema+can_por_tienda_por_sector(sec(i,j))*tiendasporruta(sec(i,j));
            % tiendSec=tiendSec+tiendasporruta(sec(i,j));
        end
    end
    %dem(i)=dema*tiendSec;
    dem(i)=dema;
    dema=0;
    tiendSec=0;
end
end

```

Figura 9-5. Función para el cálculo de demanda de cada sector.

La figura 9-5 se encarga de asignar la demanda de acuerdo al sector al que pertenezca cada uno de los vehículos.

9.4 TABLAS DE RESULTADOS DE ANÁLISIS DE VARIANZA PARA LA DENSIDAD VEHICULAR.

1. El análisis de varianza se realiza por medio de la herramienta de análisis factorial de SPSS de la siguiente manera:
 - a. Se realiza la comprobación del supuesto de homogeneidad de varianzas por medio de la prueba estadística de Levene, la cual contrasta la hipótesis nula de

que la varianza error de la variable dependiente es igual a lo largo de todos los grupos:

Tabla 9-1. Resultado prueba de Levene para la variable dependiente densidad

F	gl1	gl2	Sig.
2,996	26	54	,000

Según la prueba de Levene, cuya hipótesis nula es que las varianzas son iguales, se obtiene un valor de significancia de 0, lo cual indica que no hay evidencia suficiente para aceptar la hipótesis nula, es decir, que no se cumple el supuesto de homocedasticidad de las varianzas.

Sin embargo contando con lo robusto de la prueba ANOVA se continúa el análisis.

- b. Prueba de Varianzas o de los efectos inter-sujetos, en el software estadístico se realiza el análisis de varianzas para cada variable y buscar si existe evidencia suficiente que los valores medios de cada variable son iguales, es decir si existe o no diferencia significativa entre los factores y sus interacciones sobre la variable dependiente de densidad vehicular.

Tabla 9-2. ANOVA para los factores de día, hora y categoría en la variable densidad vehicular

Origen	Suma de cuadrados tipo III	gl	Media cuadrática	F	Sig.
Modelo corregido	46843,190 ^a	26	1801,661	63,611	,000
Intersección	67099,995	1	67099,995	2369,086	,000
Día	1846,140	2	923,070	32,591	,000
Hora	133,565	2	66,782	2,358	,104
Categoría	41563,115	2	20781,558	733,730	,000
Día * Hora	708,866	4	177,216	6,257	,000
Día * categoría	900,753	4	225,188	7,951	,000
Hora * categoría	129,952	4	32,488	1,147	,344
Día * Hora * categoría	1560,799	8	195,100	6,888	,000
Error	1529,450	54	28,323		
Total	115472,635	81			
Total corregida	48372,640	80			

a. R cuadrado = ,968 (R cuadrado corregida = ,953)

La hipótesis nula es si la media de los tratamientos de los factores afecta a la variable o no.

En este caso se ve que el valor de significancia de la variable hora es mayor que 0.05, lo cual indica que no hay diferencia significativa en los niveles del factor hora, es decir que la

hora pico de la mañana, la tarde y la noche es estadísticamente la misma, igual que la interacción entre hora y categoría. El resto de variables indican que existe diferencia significativa entre ellas.

- c. Para observar que variables son diferentes se hace la prueba de comparaciones múltiples, en este caso se usa la prueba de TUKEY y DMS (Mínima diferencia significativa), cabe notar que la variable **hora**, no requiere de hacer esta prueba puesto que en el ANOVA ya se conoce que no existe diferencia significativa, las variables **Día** y **categoría** si requieren de esta prueba para el análisis de las diferencias entre sus categorías, así se corre la prueba obteniendo los siguientes resultados:

Tabla 9-3. Comparaciones múltiples para la variable Día

Prueba	(I)Día	(J)Día	Diferencia de medias (I-J)	Error típ.	Sig.	Intervalo de confianza 95%	
						Límite inferior	Límite superior
DHS de Tukey	sábado	domingo	9,0833*	2,19878	,000	3,8244	14,3423
		jueves	-1,8367	2,19878	,682	-7,0956	3,4223
	domingo	sábado	-9,0833*	2,19878	,000	-14,3423	-3,8244
		jueves	-10,9200*	2,19878	,000	-16,1790	-5,6610
	Jueves	sábado	1,8367	2,19878	,682	-3,4223	7,0956
		domingo	10,9200*	2,19878	,000	5,6610	16,1790
DMS	Sábado	domingo	9,0833*	2,19878	,000	4,7022	13,4645
		jueves	-1,8367	2,19878	,406	-6,2178	2,5445
	domingo	sábado	-9,0833*	2,19878	,000	-13,4645	-4,7022
		jueves	-10,9200*	2,19878	,000	-15,3012	-6,5388
	Jueves	sábado	1,8367	2,19878	,406	-2,5445	6,2178
		domingo	10,9200*	2,19878	,000	6,5388	15,3012

*. La diferencia de medias es significativa al nivel ,05

Notamos tanto con TUKEY como con DMS, que existe diferencia significativa entre el día domingo con sábado y jueves y que los días jueves y sábado se pueden considerar iguales estadísticamente. Esto indica que los días feriados presentan una densidad diferente a los días de semana incluido el sábado que no hay restricción vehicular pero es igual a los días en que si hay restricción.

Tabla 9-4. Comparaciones múltiples para la variable categoría

Prueba	(I)categoría	(J)categoría	Diferencia de medias (I-J)	Error típ.	Sig.	Intervalo de confianza 95%	
						Límite inferior	Límite superior

DHS de Tukey	Alta	media	-33,1330*	1,44845	,000	-36,6237	-29,6422
		baja	21,9785*	1,44845	,000	18,4878	25,4693
	media	alta	33,1330*	1,44845	,000	29,6422	36,6237
		baja	55,1115*	1,44845	,000	51,6207	58,6022
	Baja	alta	-21,9785*	1,44845	,000	-25,4693	-18,4878
		media	-55,1115*	1,44845	,000	-58,6022	-51,6207
DMS	Alta	media	-33,1330*	1,44845	,000	-36,0369	-30,2290
		baja	21,9785*	1,44845	,000	19,0745	24,8825
	media	alta	33,1330*	1,44845	,000	30,2290	36,0369
		baja	55,1115*	1,44845	,000	52,2075	58,0155
	Baja	alta	-21,9785*	1,44845	,000	-24,8825	-19,0745
		media	-55,1115*	1,44845	,000	-58,0155	-52,2075
*. La diferencia de medias es significativa al nivel ,05.							

Según la tabla 9-4, para la variable categoría tanto con la prueba de Tukey y DMS, se tiene un valor de significancia de 0, lo cual indica que no existe evidencia suficiente para aceptar la hipótesis nula de que las calles de categoría alta, media y baja son iguales. Es decir que la categoría de las calles tiene diferencia estadística.

d. Gráficos de perfil:

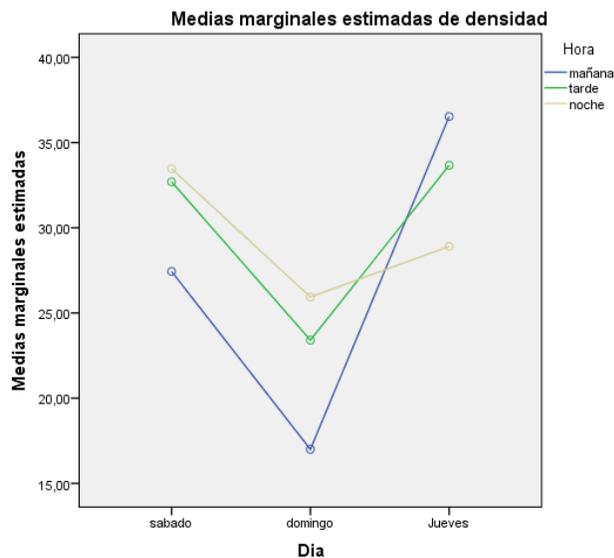


Figura 9-6. Promedio de densidad para las variables Día Vs Hora

En la figura 9-6 se indica que existe interacción entre las horas y el día y que el promedio de densidad más alto se da el día jueves en horas de la mañana. El análisis de la interacción no es de mayor importancia, ya que en el ANOVA no se cumple el supuesto de

homocedasticidad y se debe validar las respuestas con otra prueba estadística, ya sea por rangos múltiples o de tipo no paramétrica.

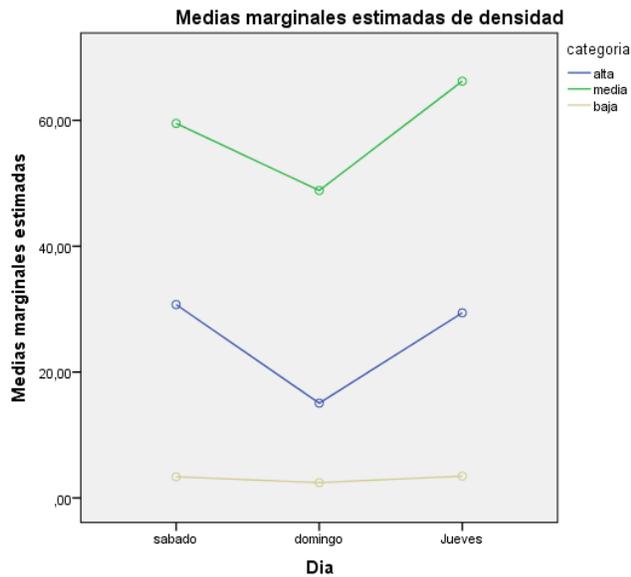


Figura 9-7.Promedio de densidad para las variables Día Vs Categoría de calle

En la figura 9-7 se muestra que la categoría con el día no tiene interacción y que el promedio de densidad más alto se da el día jueves en la calle considerada de categoría media.

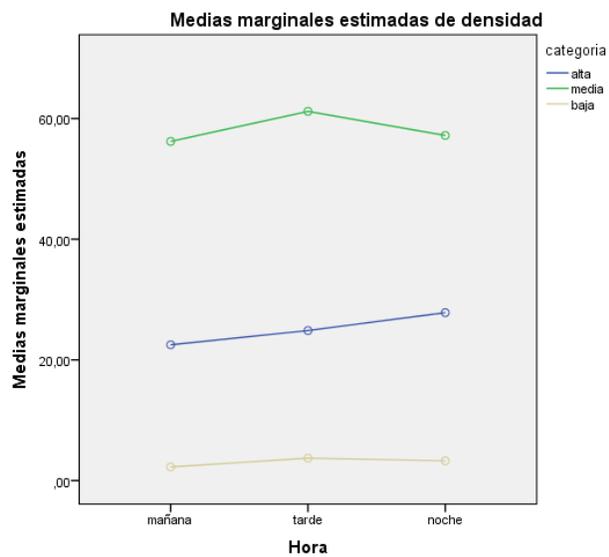


Figura 9-8.Promedio de densidad para las variables Hora Vs Categoría de calle

En la figura 9-8 se indica como la hora con la categoría no presenta interacción y el promedio de densidad más alto se da en la jornada de la tarde y en la calle considerada de categoría media.

e. Supuesto de normalidad e independencia.

Para esto se recurre a la prueba de normalidad de los residuos y los datos usando la prueba estadística de Kolmogorov Smirnov con la corrección de Lilliefors y la prueba de Shapiro- Wilk teniendo en cuenta que cuando los datos son menores a 50. Las hipótesis son:

H_0 : los datos tienen una distribución normal

H_1 : los datos no tienen una distribución normal

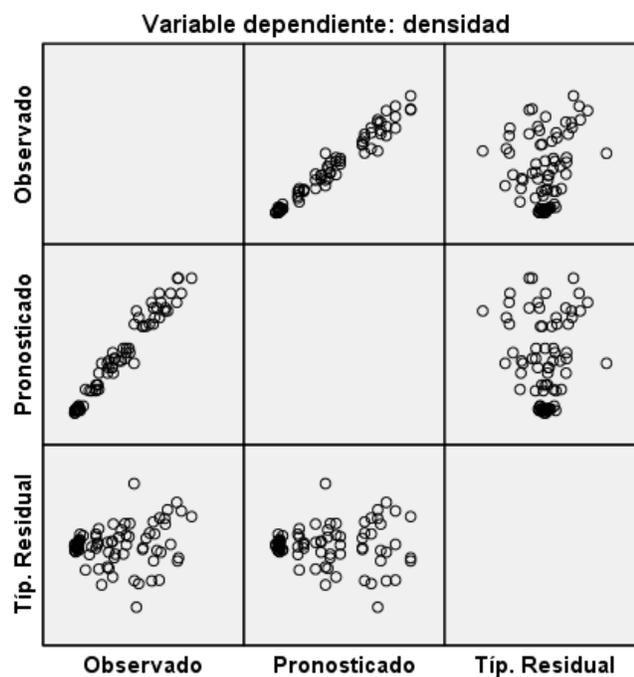
Tabla 9-5. Pruebas de normalidad

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Residuo para densidad	,134	81	,001	,955	81	,006

a. Corrección de la significación de Lilliefors

El valor de significancia en las dos pruebas indica un valor menor al valor de significancia del 5%, por lo que no hay evidencia suficiente para decir que los residuos tengan normalidad. Es decir que no se cumple el supuesto de normalidad.

f. Prueba de independencia:



Modelo\': Intersección + Dia + Hora + categoría + Dia * Hora + Dia * categoría + Hora * categoría + Dia * Hora * categoría

Figura 9-9. Prueba de independencia variable densidad vehicular

Observando la fila 3 y la columna 2 de la figura 9-9. Entre los residuos y el pronóstico, se observa que existe gran dispersión y sin comportamiento definido, por lo cual se cumple el supuesto de independencia.

- g. De acuerdo con los resultados anteriores, en los que no se cumple el supuesto de homocedasticidad, se recurre a realizar una prueba ya sea no paramétrica (Kruskal Wallis o Friedman), o por medio de la generación de rangos como lo recomienda Montgomery[56].

Usando la prueba de rangos múltiples se obtiene los siguientes resultados:

Tabla 9-6. ANOVA Pruebas de los efectos inter-sujetos para rangos de la variable densidad vehicular

Variable dependiente: Rank of densidad					
Origen	Suma de cuadrados tipo III	gl	Media cuadrática	F	Sig.

Modelo corregido	42487,167 ^a	26	1634,122	49,233	,000
Intersección	136161,000	1	136161,000	4102,303	,000
Día	1716,167	2	858,083	25,853	,000
Hora	364,796	2	182,398	5,495	,007
Categoría	38624,519	2	19312,259	581,846	,000
Día * Hora	411,315	4	102,829	3,098	,023
Día * categoría	297,481	4	74,370	2,241	,077
Hora * categoría	140,074	4	35,019	1,055	,388
Día * Hora * categoría	932,815	8	116,602	3,513	,002
Error	1792,333	54	33,191		
Total	180440,500	81			
Total corregida	44279,500	80			

a. R cuadrado = ,960 (R cuadrado corregida = ,940)

Se nota como todos los factores presentan diferencia significativa menos la interacción de día con categoría y hora categoría.

Pasando a las comparaciones múltiples se tiene:

Tabla 9-7. Comparaciones múltiples para el Factor Día

Variable dependiente: Rank of densidad							
Prueba	(I)Día	(J)Día	Diferencia de medias (I-J)	Error típ.	Sig.	Intervalo de confianza 95%	
						Límite inferior	Límite superior
DHS de Tukey	sábado	domingo	9,44444*	1,567998	,000	5,66559	13,22330
		Jueves	-,61111	1,567998	,920	-4,38996	3,16774
	domingo	sábado	-9,44444*	1,567998	,000	-13,22330	-5,66559
		Jueves	-10,05556*	1,567998	,000	-13,83441	-6,27670
	Jueves	sábado	,61111	1,567998	,920	-3,16774	4,38996
		domingo	10,05556*	1,567998	,000	6,27670	13,83441
DMS	sábado	domingo	9,44444*	1,567998	,000	6,30080	12,58809
		Jueves	-,61111	1,567998	,698	-3,75476	2,53254
	domingo	sábado	-9,44444*	1,567998	,000	-12,58809	-6,30080
		Jueves	-10,05556*	1,567998	,000	-13,19920	-6,91191
	Jueves	sábado	,61111	1,567998	,698	-2,53254	3,75476
		domingo	10,05556*	1,567998	,000	6,91191	13,19920

Basadas en las medias observadas.
El término de error es la media cuadrática (Error) = 33,191.
*. La diferencia de medias es significativa al nivel ,05.

Igual que en el ANOVA anterior se tiene diferencia significativa entre el día domingo y los otros, al igual que se corrobora como el día sábado y jueves son iguales estadísticamente en las dos pruebas (Tukey y DMS).

Tabla 9-8. Comparaciones múltiples para el Factor Hora

Variable dependiente: Rank of densidad							
Prueba	(I)Hora	(J)Hora	Diferencia de medias (I-J)	Error típ.	Sig.	Intervalo de confianza 95%	
						Límite inferior	Límite superior
DHS de Tukey	mañana	tarde	-4,57407*	1,567998	,014	-8,35293	-,79522
		noche	-4,42593*	1,567998	,018	-8,20478	-,64707
	tarde	mañana	4,57407*	1,567998	,014	,79522	8,35293
		noche	,14815	1,567998	,995	-3,63070	3,92700
	noche	mañana	4,42593*	1,567998	,018	,64707	8,20478
		tarde	-,14815	1,567998	,995	-3,92700	3,63070
DMS	mañana	tarde	-4,57407*	1,567998	,005	-7,71772	-1,43043
		noche	-4,42593*	1,567998	,007	-7,56957	-1,28228
	tarde	mañana	4,57407*	1,567998	,005	1,43043	7,71772
		noche	,14815	1,567998	,925	-2,99550	3,29180
	noche	mañana	4,42593*	1,567998	,007	1,28228	7,56957
		tarde	-,14815	1,567998	,925	-3,29180	2,99550
Basadas en las medias observadas.							
El término de error es la media cuadrática(Error) = 33,191.							
*. La diferencia de medias es significativa al nivel ,05.							

En este caso existe diferencia significativa entre la mañana con noche y tarde, además no existe diferencia significativa entre la tarde y la noche.

Tabla 9-9. Comparaciones múltiples para el Factor Categoría de calle

Variable dependiente: Rank of densidad							
Prueba	(I)categoría	(J)categoría	Diferencia de medias (I-J)	Error típ.	Sig.	Intervalo de confianza 95%	
						Límite inferior	Límite superior
DHS de Tukey	alta	media	-25,96296*	1,567998	,000	-29,74181	-22,18411
		baja	27,51852*	1,567998	,000	23,73967	31,29737
	media	alta	25,96296*	1,567998	,000	22,18411	29,74181
		baja	53,48148*	1,567998	,000	49,70263	57,26033
	baja	alta	-27,51852*	1,567998	,000	-31,29737	-23,73967
		media	-53,48148*	1,567998	,000	-57,26033	-49,70263
DMS	alta	media	-25,96296*	1,567998	,000	-29,10661	-22,81932
		baja	27,51852*	1,567998	,000	24,37487	30,66217
	media	alta	25,96296*	1,567998	,000	22,81932	29,10661
		baja	53,48148*	1,567998	,000	50,33783	56,62513
	baja	alta	-27,51852*	1,567998	,000	-30,66217	-24,37487
		media	-53,48148*	1,567998	,000	-56,62513	-50,33783
Basadas en las medias observadas.							
El término de error es la media cuadrática(Error) = 33,191.							

*. La diferencia de medias es significativa al nivel ,05.

Se observa como existe diferencia significativa en todas las categorías como es de suponer, puesto que las calles en la ciudad presentan diferentes congestiones.

2. Verificación de supuestos para la prueba con rangos múltiples.
 - a. Normalidad de los residuos:

Tabla 9-10. Pruebas de normalidad para la variable densidad con rangos múltiples

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Residuo para Rdensida	,066	81	,200*	,983	81	,339
*. Este es un límite inferior de la significación verdadera.						
a. Corrección de la significación de Lilliefors						

Según la prueba de Kolmogorov y la de Shapiro Wilk, los residuos presentan normalidad.

- b. Independencia de los residuos:

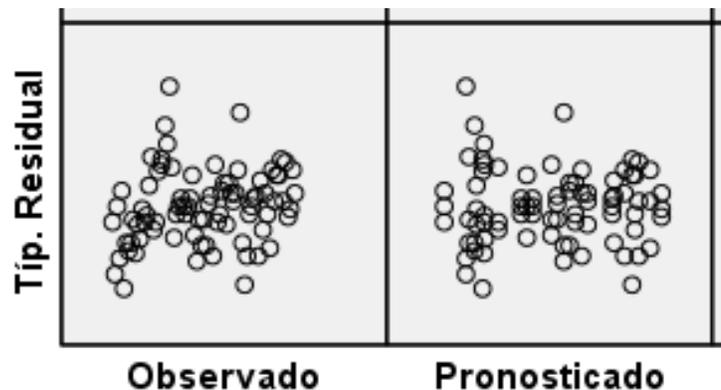


Figura 9-10. Prueba de independencia para la variable densidad vehicular con rangos múltiples

En la figura 9-10 se nota como si existe independencia en los residuos.

- c. Homocedasticidad de las varianzas. para lo cual se realiza la prueba de contraste de Levene sobre la igualdad de las varianzas, la cual contrasta la hipótesis nula de que la varianza de la variable dependiente es igual a lo largo de todos los grupos.

Tabla 9-11. Prueba de Levene para la variable dependiente densidad por rangos

F	gl1	gl2	Sig.
---	-----	-----	------

3,615	26	54	,000
-------	----	----	------

Según la prueba de Levene, no se cumple el supuesto.

- d. Ahora como se tiene un supuesto que aún no cumple se podría concluir con el ANOVA de los rangos, en el que se indica que la variable hora que en el ANOVA inicial es la misma, pero en el ANOVA de rangos es diferente la mañana de la tarde y la noche, entonces se realiza un diseño factorial completamente al azar para el factor HORA solo.
3. Tomando todo el conjunto de datos se corre el diseño para el Factor HORA con los siguientes resultados:

SUPUESTOS:

- a. Homocedasticidad de las varianzas:

Tabla 9-12. Contraste de Levene sobre la igualdad de las varianzas para la variable densidad vehicular

F	gl1	gl2	Sig.
,013	2	78	,987

Según la prueba de Levene, se tiene un valor de significancia mayor al 5%, lo cual supone varianzas constantes. Es decir que no hay evidencia para rechazar la hipótesis nula en la que las varianzas son iguales.

- b. Supuesto de Normalidad

Tabla 9-13. Pruebas de normalidad variable densidad

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Residuo para DENSIDADTOTAL	,149	81	,000	,900	81	,000

a. Corrección de la significación de Lilliefors

Según el valor de significancia no se cumple el supuesto de normalidad.

- c. Independencia:

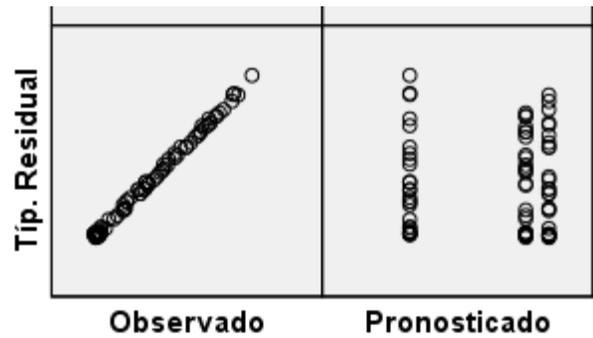


Figura 9-11. Supuesto de independencia variable densidad

Según la figura 9-11 los residuos presentan independencia.

Análisis de varianza para la variable densidad según el factor hora.

Tabla 9-14. Pruebas de los efectos inter-sujetos

Variable dependiente: DENSIDADTOTAL

Origen	Suma de cuadrados tipo III	gl	Media cuadrática	F	Sig.
Modelo corregido	133,565 ^a	2	66,782	,108	,898
Intersección	67099,995	1	67099,995	108,497	,000
HORATOTAL	133,565	2	66,782	,108	,898
Error	48239,076	78	618,450		
Total	115472,635	81			
Total corregida	48372,640	80			

a. R cuadrado = ,003 (R cuadrado corregida = -,023)

Se indica que no existe diferencia en las horas de la densidad vehicular, por lo que no es necesario hacer comparaciones múltiples.

Ahora como el supuesto de que los residuos sean normales no se cumple, recurrimos a comprobar el análisis con una prueba no paramétrica para muestras independientes según el ANOVA de Kruskal Wallis con el siguiente resultado:

Resumen de prueba de hipótesis

	Hipótesis nula	Test	Sig.	Decisión
1	La distribución de DENSIDADTOTAL es la misma entre las categorías de HORATOTAL.	Prueba Kruskal-Wallis de muestras independientes	,719	Retener la hipótesis nula.

Se muestran las significancias asintóticas. El nivel de significancia es ,05.

Figura 9-12 Resultado de ANOVA usando prueba no paramétrica de Kruskal Wallis

Como se indica, según la prueba de Kruskal Wallis, se comprueba el resultado del diseño factorial completamente al azar para la variable HORA en la que no existe diferencia significativa entre las horas pico para la densidad vehicular.

Teniendo en cuenta la clasificación anterior se puede definir diferentes calles en categorías alta, media o baja realizando una diferencia de medias para variables con muestras distintas y con varianza distinta. Para esto se hace uso del toolbox de estadísticas de matlab y la prueba T debido a la poca cantidad de datos de densidad que se tiene para las distintas calles de la ciudad de San Juan de Pasto.

La codificación en matlab se hace como se indica en la figura 9-13:

```
[Hip,p]=ttest2(x,y,0.05,'both','unequal');
Hipotesis(conref,contdat)=Hip;
pval(conref,contdat)=p;
```

Figura 9-13. Codificación prueba T para diferencia de medias con datos desbalanceados

La función **ttest2** en matlab indica que se solicita realizar la prueba de diferencia de medias entre un par de muestras por medio de la prueba *T* de *student*, para ello la función requiere tomar los datos de la primera muestra en un vector o variable *x*, y los datos de la segunda muestra en un vector o variable *y*, y los datos de la segunda muestra en un vector o variable *y*. Posteriormente la función debe tener el valor de significancia, que en este caso se da el valor del 5%, indicando que se trabajara con un nivel de confianza del 95%. En el siguiente parámetro de la función se especifica si la prueba es a dos colas o una, es decir que se contrasta las hipótesis nula y alterna así:

H₀: las medias de los datos son iguales

H_a: Las medias de los datos son distintas.

Para el caso particular se trabaja con dos colas, por lo que el valor de la función para este caso se debe especificar con el texto **both**.

Como último parámetro la función **ttest2** requiere que se indique si la cantidad de datos de las variables x,y son iguales o diferentes, es así como para el caso particular de densidad vehicular se tiene variables con cantidades diferentes de datos, por lo que se debe enviar a la función el texto **unequal**.

Si la hipótesis alterna menciona que la media de los datos es únicamente mayor o únicamente menor, entonces se estaría trabajando a una sola cola en la prueba.

Para esta prueba se tiene el valor de densidad en diferentes horas pico para 14 calles de la ciudad de San Juan de Pasto, las cuales se clasificaran de acuerdo con el resultado de la prueba T, en categorías ALTA, MEDIA y BAJA; esto por medio de la comparación de medias con las calles que se tiene como referencia en cada categoría, las cuales se tomaron con el criterio del conteo de rutas de transporte público que tiene la ciudad como se mencionó anteriormente y corresponden a los datos con los que se corre la prueba de análisis de varianza del diseño de experimentos anterior, en el que se obtuvo como resultado que existe diferencia significativa entre los tres tipos de calle..

Los datos resumidos de densidad vehicular y el nombre de las diferentes calles se indican en las tablas 9-15 a 9-17:

Usando las variables de referencia de densidad alta, media y baja para las calles denominadas "Principal Lorenzo", "Calle 15" y "capusigra": se presenta la tabla 5-16 con el valor de densidad vehicular de cada calle en unidades de Número de Vehículos/hora:

Tabla 9-15. Densidad vehicular para las calles de referencia

Principal Lorenzo Alta	Calle 15 Media	Capusigra Baja
46,65	18,27	4,52
61,57	28,53	2,37
57,95	24,35	2,71
62,40	34,13	3,20
55,88	32,51	4,12
73,73	25,38	2,90
67,17	35,78	4,55
65,20	37,84	3,10
45,02	39,73	2,77
84,32	43,33	1,22
74,93	24,36	4,89

74,25	20,30	1,16
76,97	27,80	7,07
69,36	25,25	3,87
61,24	27,54	3,89
54,42	36,95	2,79
56,96	29,98	3,32
43,59	29,19	2,97

Tabla 9-16. Calles con 8 muestras de densidad en horas pico.

CAI obrero	Hospital Infantil	Iglesia de San Sebastián	Las Margaritas	Barrio Capusigra (entrada al Éxito)	Calle 17 con AV américas	Calle 18 Av chile	Cr 22 Calle 16 y 17	Cr 24 con dos puentes	Colegio Javeriano entre carreras 24 y 25
2,935	40,869	53,745	61,600	3,840	94,227	11,917	9,749	5,590	30,622
2,573	41,093	61,015	61,172	4,242	92,804	12,540	10,851	6,714	20,839
4,151	26,860	86,088	52,785	2,310	89,067	15,607	8,339	4,296	12,390
4,084	25,020	69,660	60,471	3,462	54,175	16,433	11,761	5,725	15,935
3,182	22,502	54,357	35,984	6,560	35,218	17,333	19,844	9,525	24,656
8,024	27,462	66,355	38,300	7,455	63,030	10,043	24,578	14,397	21,879
5,067	40,133	53,448	41,970	6,824	88,889	17,395	48,484	18,916	27,378
7,529	44,082	60,597	31,012	8,587	98,217	11,033	18,824	10,976	17,251

Tabla 9-17. Calles con menos de 8 muestras de densidad vehicular en horas pico

Calle 15 Cr 26 y 27	Calle 15 Cr 25 y 26	Calle 15 Cr 23 y 24	Carrera 32 entre calles 16 a y 16 b	Carrera 32 entre calles 18 y 19	Calle 15 Cr 22 y 23	Carrera 32 entre calles 15 y 16
35,987	37,987	44,385	105,900	38,439	40,546	34,78
30,933	34,203	40,833	113,722	31,969	37,169	46,787
37,287	23,897	52,484	42,167	34,261	18,324	47,696
33,080	30,404	63,844	46,853	31,363		

Los resultados de la prueba T, son los siguientes:

El valor de significancia de cada media se muestra en la variable **pval**:

pval =

Columns 1 through 5

0.0000 0.0000 0.0000 0.0848 0.0320

```
0.0000 0.0717 0.6677 0.0230 0.8074
0.1399 0.0001 0.0024 0.0026 0.0535
```

Columns 6 through 10

```
0.1382 0.0000 0.0000 0.0000 0.0166
0.0006 0.0000 0.0535 0.0000 0.0721
0.0000 0.0082 0.0122 0.0110 0.0105
```

Columns 11 through 15

```
0.5073 0.0000 0.0000 0.0000 0.9520
0.0883 0.1127 0.0055 0.3477 0.0000
0.0300 0.0002 0.0001 0.0000 0.0000
```

Columns 16 through 17

```
NaN 0.0000
NaN 0.0000
NaN 0.0425
```

Con el comando de matlab, se genera un valor de uno o cero de acuerdo con aceptar o rechazar la hipótesis nula, este valor se indica en la variable Hipótesis:

Hipótesis =

Columns 1 through 9

```
1 1 1 0 1 0 1 1 1
1 0 0 1 0 1 1 0 1
0 1 1 1 0 1 1 1 1
```

Columns 10 through 17

```
1 0 1 1 1 0 NaN 1
0 0 0 1 0 1 NaN 1
1 1 1 1 1 1 NaN 1
```

Como se muestra en las respuestas del software matlab, se tiene tanto una variable *pval* como hipótesis con una respuesta matricial de 3 filas por 17 columnas, las cuales corresponden a las categorías ALTA, MEDIA y BAJA en las filas y a las 17 diferentes calles de la ciudad, las cuales corresponden a las columnas de las matrices respuesta. Según el valor de respuesta que genera matlab, se indica con un valor de 1 si existe suficiente evidencia para rechazar la hipótesis nula y con un valor de cero para aceptar la hipótesis nula, la cual está asignada como igualdad de medias entre las horas pico.

Las hipótesis se definen así:

Ho: las medias de los datos de densidad vehicular entre dos calles son iguales

Ha: Las medias de los datos de densidad vehicular entre dos calles son distintas.

Con estos valores se clasifica las calles de acuerdo a las medias de las calles de referencia como se indica en la tabla 9-18:

Tabla 9-18. Clasificación de calles según prueba estadística T para diferencia de medias de densidad vehicular

CALLE	HIPÓTESIS	CLASIFICACIÓN SEGÚN HIPÓTESIS	CLASIFICACIÓN SEGÚN VALOR DE SIGNIFICANCIA
CAI obrero	1 1 0	Baja	Baja
Calle 15 Cr 26 y 27	1 0 1	Media	Media
Calle 15 Cr 25 y 26	1 0 1	Media	Media
Calle 15 Cr 23 y 24	0 1 1	Alta	Alta
Calle 15 Cr 22 y 23	1 0 0	Media Baja	Media
Calle 17 con AV américas	0 1 1	Alta	Alta
Calle 18 Av chile	1 1 1	No clasifica	Baja
Cr 22 Calle 16 y 17	1 0 1	Media	Media
Cr 24 con dos puentes	1 1 1	No clasifica	Baja
Carrera 32 entre calles 15 y 16	1 0 1	Media	Media
Carrera 32 entre calles 16 a y 16 b	0 0 1	Alta Media	Alta
Carrera 32 entre calles 18 y 19	1 0 1	Media	Media
Colegio Javeriano entre carreras 24 y 25	1 1 1	No clasifica	Media
Hospital Infantil	1 0 1	Media	Media
Iglesia de San Sebastián (de la Panadería)	0 1 1	Alta	Alta
Las Margaritas	Nan Nan Nan	No clasifica	Alta
Barrio Capusigra (entrada al Éxito)	1 1 1	No clasifica	Baja

Al hacer la prueba de hipótesis de diferencia de medias con la prueba estadística T, se concluye la clasificación de las 17 calles de la ciudad, en los casos en que se clasifican en dos categorías se procede a observar el valor de significancia y clasificar según corresponda al mayor valor de significancia y la observación promedio entre las 17 calles sin tener en cuenta la referencia, como es el caso de la calle Las margaritas, la cual no se clasifica en ninguna categoría y no se encuentra el valor de significancia, por lo cual se procede a encontrar su valor promedio y ajustar su clasificación de acuerdo al promedio más cercano, que en este caso corresponde al de categoría alta.

Finalmente de acuerdo a la clasificación de las calles tomadas como referencia para las rutas de los vehículos se procede a asignar un coeficiente de costo sumado a la matriz de distancias obtenida por el algoritmo que solicita los servicios de ruta entre puntos del googlemaps. Para esto en consecuencia del análisis estadístico se da un costo mayor a las rutas en caso de pertenecer a un recorrido en las horas pico como son en la mañana de 7 a 9 am, al medio día en las horas de 12 a 2pm y en la noche en las horas de 6 a 8 pm.

Para el caso de horas no pico se toma únicamente el costo por distancia entre puntos y el valor de velocidad media.

Para el caso del día, según las respuestas estadísticas, no existe diferencia significativa entre los días de la semana y el día sábado para hacer los recorridos, por lo tanto el costo no se incrementa según el día, además cabe resaltar que el costo del día domingo o feriado no se tiene en cuenta, puesto que los vehículos no hacen recorridos estos días.

De acuerdo con la información de diferencia de medias entre las diferentes calles muestreadas, se tiene una clasificación ALTA, MEDIA y BAJA, lo cual permite generar un costo al recorrido de acuerdo al conocimiento del conductor en la ciudad, es decir que según la ruta que se genere se hace un incremento del costo dado por la vía que se transita considerando a esta con un valor según la categoría mencionada.

9.5 PROGRAMACIÓN DE SERVICIOS DE GOOGLMAPS PARA OBTENCIÓN DE MATRIZ DE DISTANCIAS

El proceso para la búsqueda de los costos se describe paso a paso según la programación realizada en el editor de texto denominado NOTEPAD ++, el cual permite escribir la programación en lenguaje HTML y JAVASCRIPT, que posteriormente se compila y ejecuta en la web.

1. Declaración de variables para adecuación de interfaz gráfica y solicitud de servicios a googlemaps.
Como ejemplo se indica la variable que permite buscar en la web de googlemaps y solicitar el servicio inicio del mapa geo referenciado.

var directionsService = new google.maps.DirectionsService();

2. Función de inicialización del mapa desde la web.
En esta función se realiza el acondicionamiento de la pantalla que se indicara al usuario, tal como el tamaño de la imagen, el renderizado, la información descrita en latitud y longitud, entre otros.

```

function initialize() {
    /*renderizacion del mapa*/
    directionsDisplay = new google.maps.DirectionsRenderer();
    /*Coordenadas en lat lon*/
    var mapDiv = new google.maps.LatLng(1.271563,-77.283754);
    var mapOptions = {
        /*Enfoque de la pantalla del mapa*/
        zoom:7,
        /*tipo de mapa por defecto ROADMAP no el satelital*/
        mapTypeId: google.maps.MapTypeId.ROADMAP,
        /*Centra el grafico a la escala pedida*/
        center: mapDiv
    }
    /*Actualizacion de mapa solicitado*/
    map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);

    /*Renderiza el nuevo mapa*/
    directionsDisplay.setMap(map);
    /*Inicializa la funcion que procesa todas las coordenadas*/
    google.maps.event.addListenerOnce(map, 'tilesloaded', recorrido);
}

```

Figura 9-14 Función de inicialización de API de google.

3. Se asigna en dos variables tipo vector las coordenadas en latitud y longitud para el origen y destino de todos los puntos que se requiere obtener la distancia entre ellos.

```

var inicio = new Array();
var destino = new Array();
var distancia = "";

var start="";
var end="";
..
inicio[152] = '1.218361,-77.290987';
inicio[153] = '1.210573,-77.275184';

*/
destino[0] = '1.24788,-77.265499';
destino[1] = '1.202142,-77.284454';
destino[2] = '1.200598,-77.282517';
destino[3] = '1.200018,-77.281118';

```

Figura 9-15. Variables de inicio y destino de los sectores a visitar.

4. Con los datos de inicio y destino se realiza una funcion que genere el recorrido, en este caso se realiza un llamado a calcular distancia entre los puntos de forma que se envíe el primer inicio y la combinacion con todos los destinos, es decir la primera fila de la matriz de distancias, esto se realiza para todos los puntos hasta obtener la matriz completa.

```

function recorrido( )
{
    /*Comprobar que el vector inicio este lleno*/
    if(inicio.length >=1)
    {
        /*variables para coordenadas de inicio y destino*/
        start=inicio[i];
        end = destino[j];

        /*Comprobar que el vector destino este lleno*/
        if(destino.length>=1){

            /*Incrementa el i en 1 para tomar los j desde 0*/
            if(destino.length==j)
            {
                i +=1;
                j=0;
                /*Imprime un separador de contador i*/
                document.getElementById('geo').innerHTML += '-----'+ "\n";
            }
            /*llamar a la funcion recorrido cada 2.5 segundos*/
            tempo = setTimeout("recorrido()", 2500);

            /*LLamada a la funcion que dibuja la ruta minima que tiene dos parametro inicio destino*/
            calcularRuta(start,end);
            /*incrementa j de destino*/
            j+=1;
        }
    }
}

```

Figura 9-16 Función Recorrido. Solicitud de servicio de googlemaps para realizar el recorrido de un punto origen hacia un destino en coordenadas de latitud y longitud

Es importante recalcar en esta funcion que se debe dar un tiempo de espera para el llamado a la funcion que calcule la distancia entre puntos puesto que en la red el tiempo de respuesta de la aplicación de googlemaps tarda cerca de 2 segundos en generar el valor y trazado de la ruta entre dos puntos. Para el caso particular se da un tiempo de espera de 2,5 segundos que garantice que la red alcanza a generar la ruta y el calculo de distancia que depende de googlemaps.

5. En seguida se procede a calcular la ruta entre dos puntos, esto despues de tener el valor de coordenadas en latitud y longitud de los dos puntos llamados inicio y destino.

```

function calcularRuta(ini, fin) {
    var request = {
        origin: ini,
        destination: fin,
        travelMode: google.maps.DirectionsTravelMode.DRIVING
    };

    /*Llama al servicio y pide una respuesta*/
    directionsService.route(request, function(response, status) {
        /*Si encuentra una ruta*/
        if (status == google.maps.DirectionsStatus.OK) {
            /*Envia el recorrido */
            directionsDisplay.setDirections(response);

            /*Llama al metodo para calcular la distancia que lleva como parametro la ruta*/
            calcularDistancia(directionsDisplay.directions);

            /*Imprime los datos de las coordenadas a las que se le calcula la ruta y distancia*/
            // document.getElementById('geo').innerHTML += 'DE -> '+start+' A -> '+end+' DISTANCIA
            document.getElementById('geo').innerHTML += distancia + "\n";
        }
    });
}

```

Figura 9-17 Función calcular ruta.

En esta función es de resaltar la solicitud a la aplicación de googlemaps para determinar la ruta entre los dos puntos por medio de la instrucción ***directionsService.route(request,function(response,status))***.

Esta instrucción hace un llamado al aplicativo de googlemaps y este se encarga de entregar la ruta mínima georeferenciada.

6. Cálculo de distancia de la ruta. Para esto se hace el llamado a la función que calcule la distancia real de la ruta anterior entre dos puntos, para ello se hace uso de la instrucción de googlemaps. ***myroute.legs.length***

```

function calcularDistancia(result) {
    var total = 0;
    var tipoUnidades=" km";

    var myroute = result.routes[0];
    for (var i = 0; i < myroute.legs.length; i++) {
        total += myroute.legs[i].distance.value;
    }
    //distancia = (total/1000).toFixed(2) + tipoUnidades;
    distancia = (total/1000).toFixed(4);
}

/*Llama a la función que inicializa todos los servicios de google maps*/
google.maps.event.addDomListener(window, 'load', initialize);

```

Figura 9-18 Función calcular distancia.

Por medio de las instrucciones de googlemaps se realiza la adecuación de la distancia de la ruta en unidades de kilómetros y con la precisión que se desee, que en este caso es con 4 puntos decimales.

- Finalmente se toma la distancia de cada punto y se muestra en la pantalla del usuario.

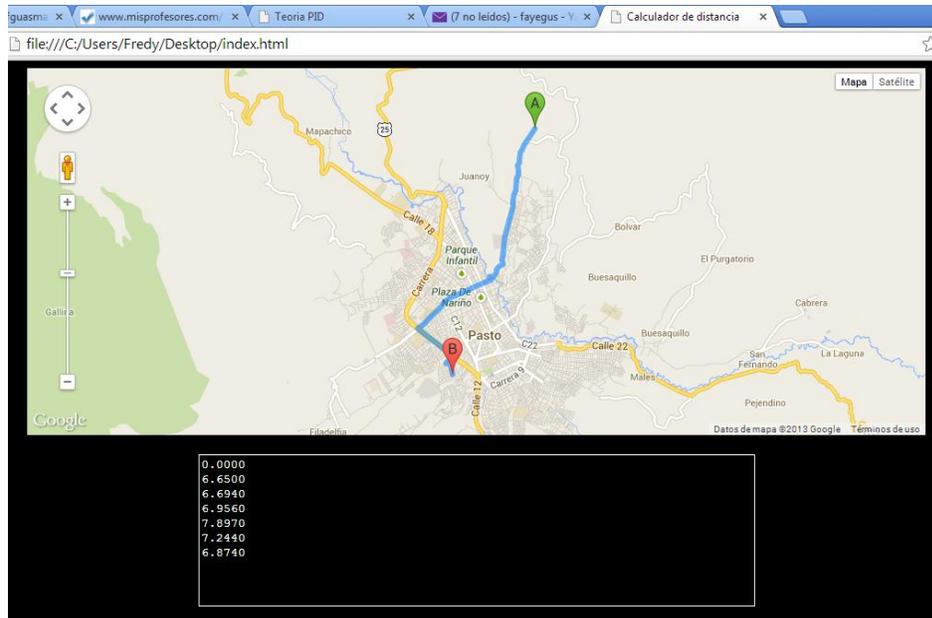


Figura 9-19. Aplicativo para calcular distancia entre puntos.

De esta forma se obtiene la matriz de distancias para los 154 sectores de la ciudad de San Juan de Pasto en el orden que se indica en las coordenadas de puntos para todos los vehículos. (ver tabla 9-20)

	A	B	C	EU	EV	EW	EX
1	0,0000	12,6490	12,7650	...	9,9670	10,4930	11,1200
2	12,2940	0,0000	0,3760	...	3,3260	2,6240	2,8390
3	12,3300	0,3760	0,0000	...	3,3620	2,6600	2,6210
4	12,5810	0,6520	0,2670	...	3,6130	2,9110	2,4850
152	9,9200	4,0130	4,1290	...	0,0000	1,0640	3,3680
153	10,1960	2,9440	3,0600	...	0,8990	0,0000	3,6440
154	8,1400	3,4240	3,0590	...	3,4930	2,8960	0,0000

Figura 9-20. Matriz de distancias de 154x154 puntos.

Los datos que se muestran en la figura 9-20, corresponde a la matriz de distancias, en la que se observa como existe diferencia entre los puntos de ida con los de regreso, es decir que la matriz no es simetrica, aunque la diferencia en la simetria es baja, lo cual permite resolverse el VRP simetrico con un factor de error pequeño, aunque no es el caso de este proyecto en el cual se resuelve el problema con matriz no simetrica.

9.6 TABLAS DE DATOS PARA CARACTERIZACIÓN DE LA RESTRICCIÓN DE DEPENDENCIA DEL TIEMPO.

En la tabla 9-1 se resume el valor de la ruta que tiene cada vehículo, el costo por combustible, la distancia, el tiempo de recorrido según la aplicación de googlemaps, la velocidad media del recorrido y el tiempo real que se tarde el vehículo según la información suministrada por el propietario del vehículo, además de la carga que entrega cada ruta.

Tabla 9-19. Resumen información de rutas de vehículos segun entrevista y aplicacion de google maps

Vehículo	Costo combustible CP\$	Distancia ruta Km	Tiempo Ruta Hr min	Tiempo real Hr	Velocidad media Km/h	No de sectores	Demanda No de canastos	No de tiendas
1	15000	22,3	1h 37 min	9	13,79	12	60	120
2	15000	23,8	1h 36 min	9	14,88	12	52	120
3	11000	15,6	1h 12 min	5,5	13,00	10	35	70
4	17500	19	1h 22min	7,5	13,90	10	70	220
5	30000	44,5	2h 25min	9,5	18,41	12	40	50
6	24000	21,8	1h 39min	7,5	13,21	14	86	120
7	10000	20,8	1h 27min	7,5	14,34	15	82	200
8	15000	34,4	2h 29min	7,5	13,85	16	40	180
9	12000	17,9	1h 24min	9,5	12,79	9	43	80
10	15000	21,4	1h 34min	7	13,66	11	25	70
11	12000	16,4	1h 9min	6	14,26	9	71	135
12	15000	18,4	1h 21min	10	13,63	14	50	120
13	15000	24,8	1h 51min	7	13,41	10	28	78
14	15000	22,6	1h 46min	8	12,79	13	50	86
15	18000	22	1h 36min	9	13,75	10	58	150
16	15000	14	1h 7min	7	12,54	10	38	60
17	15000	23,4	1h 29min	8	15,78	10	38	60
18	15000	14,4	1h 5min	7	13,29	8	35	60
19	10000	10,3	0h 52min	5	11,88	6	28	70
20	15000	14,5	1h 6min	8	13,18	11	45	50
21	15000	15	1h 11min	8	12,68	8	53	100
22	17000	16,4	1h 7min	7	14,69	7	47	150
23	12000	17,4	1h 11min	6	14,70	6	40	80
24	30000	25,9	1h 46min	9,5	14,66	11	80	110
Valor promedio	15979,17	20,71	1,47 h	7,71	13,88	10,58	49,75	105,79
Desviación estándar	5192,97	7,16	0,40	1,33	1,31	2,64	17,38	47,50

Tabla 9-20. Coordenadas con puntos de todos los vehículos:

Índice	Latitud, longitud	Lugar
1	1.24788,-77.265499	Planta Colácteos Aranda
2	1.202142,-77.284454	Caicedo
3	1.200598,-77.282517	Granada
4	1.200018,-77.281118	ciudad jardín
5	1.199182,-77.282492	Nueva Colombia

6	1.197626,-77.277115	María Isabel (Estadio libertad)
7	1.200362,-77.283359	Chapál.
8	1.20107,-77.270046	Las Violetas
9	1.205596,-77.277342	Aire libre
10	1.207806,-77.279598	Av Las Américas
11	1.204309,-77.280475	Atahualpa
12	1.204845,-77.280453	San Miguel
13	1.194655,-77.27646	Pilar Cr5 12 C-49 El Pilar, Pasto, Nariño, Colombia
14	1.195749,-77.274295	Santa clara
15	1.196125,-77.272383	San Martin
16	1.1968,-77.270368	Mercado el potrерillo Calle 15D # 5B-1 a 5B-99
17	1.196886,-77.273522	Progreso (calle 14 7 20)
18	1.205746,-77.274491	Centro comercial el tigre de la rebaja cede champagnat.
19	1.20536,-77.272903	Fátima Calle 19 # 14-2 a 14-100
20	1.193583,-77.272535	La rosa Carrera 4 # 13-2 a 13-100
21	1.18927,-77.273811	Chambú
22	1.188198,-77.273504	Altos de chapál
23	1.191716,-77.270157	La minga.
24	1.215143,-77.274433	Los dos puentes
Índice	Latitud, longitud	Lugar
25	1.22012,-77.280414	Las cuadras
26	1.218243,-77.28528	San Ignacio.
27	1.217288,-77.283632	Bombona.
28	1.209951,-77.285435	Obrero
29	1.209608,-77.279555	Centro (Av Boyacá, Av Las américas)
30	1.210809,-77.282683	Santiago
31	1.215389,-77.280409	Centro (Calle 16 hasta Kr 30). Iglesia la Catedral
32	1.19901,-77.261016	Villa flor
33	1.201627,-77.25445	Caicedonia (Cerca al barrio popular)
34	1.203129,-77.263489	La esmeralda (Cerca al parque Bolívar)
35	1.205103,-77.26123	Mercedario
36	1.207044,-77.262159	Pucalpa 2.
37	1.212053,-77.263119	Los sauces
38	1.21038,-77.259124	La florida
39	1.209114,-77.259704	Carolina
40	1.214295,-77.264363	Carlos Pizarro, San Ezequiel, Simón Bolívar. (Cerca al ancianato)
41	1.188756,-77.238964	Mocondino
42	1.204459,-77.210125	La laguna
43	1.212697,-77.2175	Cabrera

44	1.204845,-77.225482	San Fernando
45	1.204802,-77.268483	Supermercado metrópoli (Av Colombia, hospital departamental).
46	1.198002,-77.264519	Macro económico. (Lorenzo)
47	1.204652,-77.267428	Supermercado Líder.
48	1.203757,-77.292633	Merca Feliz (Tamasagra).
49	1.204502,-77.293154	Tamasagra.
50	1.194677,-77.263934	Lorenzo, villa docente
51	1.221493,-77.282404	Maridiaz
52	1.222994,-77.286976	La aurora
53	1.212751,-77.286567	Crecebillas. Colegio normalista de Pasto
54	1.215872,-77.29213	San Vicente
55	1.214714,-77.296764	Panorámico
56	1.213856,-77.299726	Gualcaloma
57	1.210466,-77.299339	Anganoy (Alcaldía)
58	1.211968,-77.298996	Los Rosales
59	1.213384,-77.301142	Colon
60	1.214059,-77.285172	San Felipe
61	1.221278,-77.285435	Calle 16 salida al Hospital San Pedro
62	1.194505,-77.265572	Lorenzo
63	1.196457,-77.268265	Madrigal
Índice	Latitud, longitud	Lugar
64	1.194955,-77.26893	Cantarana Bajo
65	1.195964,-77.269081	Cantarana alto
66	1.194762,-77.269166	Emilio Botero
67	1.193003,-77.265175	Miraflores
68	1.192445,-77.269252	12 De octubre
69	1.190922,-77.267815	Belén
70	1.191287,-77.265304	Porvenir
71	1.189598,-77.26256	Rosario
72	1.191502,-77.263738	Triunfo
73	1.193068,-77.263073	villa olímpica
74	1.179745,-77.254671	Jamondino
75	1.191223,-77.268716	Belén
76	1.203097,-77.268435	Chile
77	1.198013,-77.268736	Terminal
78	1.192987,-77.261976	Villa del sol
79	1.192612,-77.261353	Arnulfo Guerrero
80	1.205521,-77.269615	Parque Bolívar
81	1.206948,-77.27153	Avenida Colombia
82	1.207956,-77.273522	Navarrete
83	1.207119,-77.271033	El prado
84	1.226877,-77.283305	Palermo

85	1.230374,-77.286676	Pandiaco
86	1.225451,-77.288477	El dorado
87	1.224346,-77.290333	Santa Ana
88	1.225762,-77.290065	La colina
89	1.224303,-77.291654	San Pedro
90	1.223252,-77.291268	San Rafael
91	1.230546,-77.268698	Nueva aranda
92	1.227242,-77.267925	San Alvano
93	1.22381,-77.267539	La esperanza
94	1.205403,-77.263591	Parque Baviera
95	1.206283,-77.264532	El ejido
96	1.203837,-77.271505	El recuerdo
97	1.204191,-77.273317	Fátima
98	1.200691,-77.269896	Venecia
99	1.203719,-77.276578	Champagnat
100	1.222973,-77.267922	Villa nueva
101	1.226577,-77.267879	Nuevo horizonte
102	1.228637,-77.271827	Villa del norte
103	1.233195,-77.270742	Nuevo sol
104	1.196093,-77.27035	Potrerrillo
105	1.218961,-77.253506	Buesaquillo
106	1.212204,-77.272299	Corporación
Índice	Latitud, longitud	Lugar
107	1.205231,-77.288739	Sumatambo
108	1.206175,-77.292945	Mijitayo
109	1.201842,-77.287709	Agualongo
110	1.204245,-77.285606	Bachué (Frente al éxito)
111	1.212461,-77.2922	El vergel
112	1.208878,-77.292387	El bosque
113	1.21038,-77.292988	Villa Sofía
114	1.217438,-77.292233	Villa Aurora
115	1.208063,-77.281848	Avenida Boyacá
116	1.196393,-77.282688	Fundadores (Niza)
117	1.196221,-77.285392	Quito López (Vía a Jongovito)
118	1.199847,-77.285474	Los guaduales. (Jardín de las mercedes)
119	1.205703,-77.284251	Álamos (Los balcones)
120	1.224882,-77.27378	Cementerio central
121	1.214982,-77.278982	Centro Parque Nariño
122	1.210294,-77.279899	Libertad Centro
123	1.230632,-77.267803	Floresta (Aranda)
124	1.234407,-77.265314	Simón Bolívar (Aranda).
125	1.232176,-77.270979	Sindagua (Aranda)
126	1.223767,-77.283117	Universidad Mariana

127	1.232391,-77.293202	Universidad de Nariño
128	1.230503,-77.292983	Universidad Cooperativa de Colombia
129	1.227156,-77.281787	Avenida los estudiantes
130	1.226942,-77.29449	Altos de la colina
131	1.230975,-77.293295	Quintas de San Pedro
132	1.226813,-77.296421	Laureles (altos de la colina)
133	1.211002,-77.280088	Centro (San Andresito)
134	1.219176,-77.281676	Parque infantil
135	1.216237,-77.27908	Centro (Éxito del centro)
136	1.2157,-77.280024	Misterpollo (Centro)
137	1.215572,-77.280843	Pollo sorpresa (centro)
138	1.223767,-77.270121	Corazón de Jesús
139	1.221278,-77.268747	Belarcazar (Corazón de Jesús-hospital civil)
140	1.226942,-77.267546	La lomita
141	1.216484,-77.270967	Aquines
142	1.210241,-77.273553	Carrera 19 con calle 20
143	1.219069,-77.269596	Hospital civil
144	1.219262,-77.266205	Alameda.
145	1.198195,-77.291271	Jerusalén, la cruz (Tamasagra)
146	1.197047,-77.293004	Las palmas (Tamasagra)
Índice	Latitud, longitud	Lugar
147	1.197079,-77.272646	San Martin (sector potrerillo)
148	1.197122,-77.273676	Santa clara (sector potrerillo)
149	1.20772,-77.277495	Avenida las Américas con Julian Bucheli
150	1.208664,-77.288181	Capusigra (Éxito panamericana)
151	1.218833,-77.288997	La castellana
152	1.224539,-77.291407	San Rafael
153	1.218361,-77.290987	Marilúz 1, 2,3
154	1.210573,-77.275184	Morasurco

Los barrios y sus nombres en la ciudad de Pasto, se pueden corroborar en la siguiente dirección:

<http://www.pasto.gov.co/index.php/comunas-barrios-corregimientos-veredas>

Tabla 9-21. Rutas que atienden a cada sector y demanda de cada sector

Sector	Rutas que lo atienden	Demanda
2	1 15	11,898
3	1	5,454
4	1	5,454
5	1	5,454

6	1	5,454
7	1 15	11,898
8	1	5,454
9	1 13 10	11,065
10	1	5,454
11	1	5,454
12	1	5,454
13	2 22	12,560
14	2 8	7,393
15	2	4,727
16	22 2	12,560
17	2	4,727
18	2	4,727
19	2	4,727
20	2	4,727
21	22 2	12,560
22	22 2	12,560
23	8 2	7,393
24	3 18 20 2	18,116
25	19 3 18 18 3 10	25,878
26	24 6 3 9 18	28,880
27	9 13 6 3 18 14	28,158
28	23 3	11,889
29	3	3,889
30	23 14 18 3	21,056
31	3	3,889
32	4 10 10 8	15,442
33	4	7,777
34	4	7,777
35	4	7,777
36	4	7,777
37	4	7,777
38	13 4	10,888
39	10 13 4	13,388
40	4	7,777
41	5	3,637
42	5	3,637
43	5	3,637
44	5	3,637
45	5	3,637
46	5	3,637
47	5	3,637
48	5	3,637
49	14 21 5	15,374

50	10 8 5 7	14,660
51	6 11	15,491
52	6 9 14	16,158
53	6 9	11,991
54	6 9 24	19,991
55	6 9	11,991
56	6	6,616
57	9 6	11,991
58	6	6,616
59	6	6,616
60	9 6	11,991
61	6	6,616
62	7 7	11,716
63	7	5,858
64	7 8	8,524
65	7	5,858
66	7	5,858
67	8 7 5	12,160
68	7	5,858
69	7	5,858
70	7	5,858
71	8 7	8,524
72	7	5,858
73	8 7	8,524
74	8	2,666
75	8	2,666
76	8	2,666
77	8	2,666
78	8	2,666
79	8	2,666
80	13 10	5,611
81	10	2,500
82	23 10	10,500
83	10	2,500
84	11	8,875
85	11 17 24	21,097
86	11	8,875
87	11	8,875
88	24 11 17	21,097
89	11	8,875
90	11	8,875
91	12 16	8,068
92	12	3,846
93	12	3,846
94	12	3,846
95	12	3,846

96	13 12	6,957
97	12	3,846
98	12	3,846
99	12	3,846
100	12	3,846
101	16 12	8,068
102	12	3,846
103	16 12	8,068
104	13	3,111
105	13	3,111
106	13	3,111
107	14 21	11,738
108	15 14 8	13,277
109	15 14 21	18,182
110	21 14	11,738
111	24 14	12,167
112	14	4,167
113	14	4,167
114	14	4,167
115	15	6,444
116	15	6,444
117	15	6,444
118	15	6,444
119	15	6,444
120	16	4,222
121	16	4,222
122	16	4,222
123	20 16	8,722
124	16	4,222
125	16	4,222
126	17	4,222
127	17	4,222
128	17	4,222
129	17	4,222
130	17 24	12,223
131	17	4,222
132	17	4,222
133	18	5,000
134	19	5,600
135	19	5,600
136	19	5,600
137	19	5,600
138	20	4,500
139	20	4,500
140	20	4,500
141	20	4,500

142	20	4,500
143	20	4,500
144	20	4,500
145	21 21 20	19,642
146	21	7,571
147	22	7,833
148	22	7,833
149	23	8,000
150	23	8,000
151	24	8,000
152	24	8,000
153	24	8,000
154	24	8,000

9.7 TABLAS DE DENSIDAD VEHICULAR

Las tablas indicadas a continuación, muestran los valores de densidad vehicular para las diferentes calles de la ciudad de San Juan de Pasto, en los días jueves, sábado y domingo en las horas pico para las jornadas de mañana, tarde y noche.

Tabla 9-22. Densidad vehicular calle principal Lorenzo día sábado.

Total vehículos Aforados en el día: 3711			
Calle principal Lorenzo Día sábado		Categoría	Alta
JORNADA	DENSIDADES TOTALES		
	HORA	DENSIDAD (Veh/km)	
Mañana	07:00:00 a.m.	46,65	
	08:00:00 a.m.	61,57	
	09:00:00 a.m.	57,95	
Medio Día	12:00:00 p.m.	62,40	
	01:00:00 p.m.	55,88	
	02:00:00 p.m.	73,73	
Tarde/Noche	06:00:00 p.m.	67,17	
	07:00:00 p.m.	65,20	
	08:00:00 p.m.	45,02	

Tabla 9-23. Densidad vehicular calle principal Lorenzo día domingo

Total vehículos Aforados en el día: 939			
Calle principal Lorenzo día Domingo		Categoría	Alta
JORNADA	DENSIDADES TOTALES		

	HORA	DENSIDAD (Veh/km)
Mañana	07:00:00 a.m.	28,15
	08:00:00 a.m.	37,70
	09:00:00 a.m.	40,32
Medio Día	12:00:00 p.m.	51,93
	01:00:00 p.m.	49,40
	02:00:00 p.m.	49,76
Tarde/Noche	06:00:00 p.m.	57,24
	07:00:00 p.m.	65,58
	08:00:00 p.m.	59,62

Tabla 9-24. Densidad vehicular calle principal Lorenzo día jueves

Total vehículos Aforados en el día: 4014			
Calle principal Lorenzo día Jueves		Categoría	Alta
JORNADA	DENSIDADES TOTALES		
	HORA	DENSIDAD (Veh/km)	
Mañana	07:00:00 a.m.	84,32	
	08:00:00 a.m.	74,93	
	09:00:00 a.m.	74,25	
Medio Día	12:00:00 p.m.	76,97	
	01:00:00 p.m.	69,36	
	02:00:00 p.m.	61,24	
Tarde/Noche	06:00:00 p.m.	54,42	
	07:00:00 p.m.	56,96	
	08:00:00 p.m.	43,59	

Tabla 9-25. Densidad vehicular calle 15 día sábado

Total vehículos Aforados en el día: 1865			
Calle 15 Día sábado		Categoría	Media
JORNADA	DENSIDADES TOTALES		
	HORA	DENSIDAD (Veh/km)	
Mañana	07:00:00 a.m.	18,27	
	08:00:00 a.m.	28,53	
	09:00:00 a.m.	24,35	
Medio Día	12:00:00 p.m.	34,13	
	01:00:00 p.m.	32,51	
	02:00:00 p.m.	25,38	
Tarde/Noche	06:00:00 p.m.	35,78	
	07:00:00 p.m.	37,84	
	08:00:00 p.m.	39,73	

Tabla 9-26. Densidad vehicular calle 15 día domingo

Total vehículos Aforados en el día: 939			
Calle 15 día Domingo		Categoría	Media
JORNADA	DENSIDADES TOTALES		
	HORA	DENSIDAD (Veh/km)	
Mañana	07:00:00 a.m.	8,80	
	08:00:00 a.m.	16,37	
	09:00:00 a.m.	18,18	
Medio Día	12:00:00 p.m.	17,60	
	01:00:00 p.m.	17,18	
	02:00:00 p.m.	16,38	
Tarde/Noche	06:00:00 p.m.	16,17	
	07:00:00 p.m.	11,53	
	08:00:00 p.m.	13,29	

Tabla 9-27. Densidad vehicular calle 15 día jueves

Total vehículos Aforados en el día: 1751			
Calle 15 día Jueves		Categoría	Media
JORNADA	DENSIDADES TOTALES		
	HORA	DENSIDAD (Veh/km)	
Mañana	07:00:00 a.m.	43,33	
	08:00:00 a.m.	24,36	
	09:00:00 a.m.	20,30	
Medio Día	12:00:00 p.m.	27,80	
	01:00:00 p.m.	25,25	
	02:00:00 p.m.	27,54	
Tarde/Noche	06:00:00 p.m.	36,95	
	07:00:00 p.m.	29,98	
	08:00:00 p.m.	29,19	

Tabla 9-28. Densidad vehicular Capusigra día Sábado

Total vehículos Aforados en el día: 305			
Capusigra Día sábado		Categoría	Baja
JORNADA	DENSIDADES TOTALES		
	HORA	DENSIDAD (Veh/km)	
Mañana	07:00:00 a.m.	4,52	
	08:00:00 a.m.	2,37	
	09:00:00 a.m.	2,71	
Medio Día	12:00:00 p.m.	3,20	
	01:00:00 p.m.	4,12	

	02:00:00 p.m.	2,90
Tarde/Noche	06:00:00 p.m.	4,55
	07:00:00 p.m.	3,10
	08:00:00 p.m.	2,77

Tabla 9-29. Densidad vehicular Capusigra día domingo

Total vehículos Aforados en el día: 207		
Capusigra día Domingo		Baja
JORNADA	DENSIDADES TOTALES	
	HORA	DENSIDAD (Veh/km)
Mañana	07:00:00 a.m.	1,17
	08:00:00 a.m.	1,41
	09:00:00 a.m.	0,89
Medio Día	12:00:00 p.m.	3,43
	01:00:00 p.m.	2,76
	02:00:00 p.m.	2,23
Tarde/Noche	06:00:00 p.m.	4,05
	07:00:00 p.m.	4,12
	08:00:00 p.m.	1,82

Tabla 9-30. Densidad vehicular Capusigra día jueves

Total vehículos Aforados en el día: 250		
Capusigra día Jueves		Baja
JORNADA	DENSIDADES TOTALES	
	HORA	DENSIDAD (Veh/km)
Mañana	07:00:00 a.m.	1,22
	08:00:00 a.m.	4,89
	09:00:00 a.m.	1,16
Medio Día	12:00:00 p.m.	7,07
	01:00:00 p.m.	3,87
	02:00:00 p.m.	3,89
Tarde/Noche	06:00:00 p.m.	2,79
	07:00:00 p.m.	3,32
	08:00:00 p.m.	2,97

Los datos indicados en las tablas anteriores se obtienen por aforo directo sobre las vías consideradas de alta densidad vehicular, media y baja de acuerdo con el número total de rutas de buses que pasan por las vías que se tienen como referencia, así la calle 15 por ejemplo tiene 11 rutas de buses que transitan por allí, mientras la calle 18 presenta 9 rutas y la calle del barrio Capusigra no tiene rutas.

Para la toma de datos se debe establecer dos puntos de referencia con visibilidad completa en la calle y se mide la distancia entre ellos.

Se hace el conteo de vehículos desde el punto de inicio al punto final junto con el tiempo que viaja cada vehículo en este tramo.

Con estos valores se calcula la velocidad media.

Finalmente se encuentra el valor de la densidad vehicular como la razón entre el número de vehículos por hora sobre la velocidad promedio.

Densidad Vehicular=Número de vehículos por hora/Velocidad media

La variable densidad contiene el valor de todos los datos en cada categoría.

La tabla 9-31 indica los valores que se ingresan al software estadístico así:

Tabla 9-31. Variables categóricas y densidad vehicular

Día	Hora	Categoría	Densidad
1,00	1,00	1,00	18,27
1,00	1,00	1,00	28,53
1,00	1,00	1,00	24,35
1,00	2,00	1,00	34,13
1,00	2,00	1,00	32,51
1,00	2,00	1,00	25,38
1,00	3,00	1,00	35,78
1,00	3,00	1,00	37,84
1,00	3,00	1,00	39,73
1,00	1,00	2,00	46,65
1,00	1,00	2,00	61,57
1,00	1,00	2,00	57,95
1,00	2,00	2,00	62,40
1,00	2,00	2,00	55,88
1,00	2,00	2,00	73,73
1,00	3,00	2,00	67,17
1,00	3,00	2,00	65,20
1,00	3,00	2,00	45,02
1,00	1,00	3,00	4,52
1,00	1,00	3,00	2,37
1,00	1,00	3,00	2,71
1,00	2,00	3,00	3,20
1,00	2,00	3,00	4,12
1,00	2,00	3,00	2,90

1,00	3,00	3,00	4,55
1,00	3,00	3,00	3,10
1,00	3,00	3,00	2,77
2,00	1,00	1,00	8,80
2,00	1,00	1,00	16,37
2,00	1,00	1,00	18,18
2,00	2,00	1,00	17,60
2,00	2,00	1,00	17,18
2,00	2,00	1,00	16,38
2,00	3,00	1,00	16,17
2,00	3,00	1,00	11,53
2,00	3,00	1,00	13,29
2,00	1,00	2,00	28,15
2,00	1,00	2,00	37,70
2,00	1,00	2,00	40,32
2,00	2,00	2,00	51,93
2,00	2,00	2,00	49,40
2,00	2,00	2,00	49,76
2,00	3,00	2,00	57,24
2,00	3,00	2,00	65,58
2,00	3,00	2,00	59,62
2,00	1,00	3,00	1,17
2,00	1,00	3,00	1,41
2,00	1,00	3,00	,89
2,00	2,00	3,00	3,43
2,00	2,00	3,00	2,76
2,00	2,00	3,00	2,23
2,00	3,00	3,00	4,05
2,00	3,00	3,00	4,12
2,00	3,00	3,00	1,82
3,00	1,00	1,00	43,33
3,00	1,00	1,00	24,36
3,00	1,00	1,00	20,30
3,00	2,00	1,00	27,80
3,00	2,00	1,00	25,25
3,00	2,00	1,00	27,54
3,00	3,00	1,00	36,95
3,00	3,00	1,00	29,98
3,00	3,00	1,00	29,19
3,00	1,00	2,00	84,32
3,00	1,00	2,00	74,93
3,00	1,00	2,00	74,25
3,00	2,00	2,00	76,97
3,00	2,00	2,00	69,36
3,00	2,00	2,00	61,24
3,00	3,00	2,00	54,42

3,00	3,00	2,00	56,96
3,00	3,00	2,00	43,59
3,00	1,00	3,00	1,22
3,00	1,00	3,00	4,89
3,00	1,00	3,00	1,16
3,00	2,00	3,00	7,07
3,00	2,00	3,00	3,87
3,00	2,00	3,00	3,89
3,00	3,00	3,00	2,79
3,00	3,00	3,00	3,32
3,00	3,00	3,00	2,97

9.8 TABLAS DE COORDENADAS EN LATITUD Y LONGITUD PARA LAS RUTAS DE LOS 24 VEHÍCULOS DE COLÁCTEOS POR SECTORES

Tabla 9-32. Coordenadas en latitud y longitud para las 6 primeras rutas

Ruta 1	Ruta 2	Ruta 3	Ruta 4	Ruta 5	Ruta 6
1.24788,-77.265499	1.24788,-77.265499	1.24788,-77.265499	1.24788,-77.265499	1.24788,-77.265499	1.24788,-77.265499
1.2038370,-77.2715073	1.2305460,-77.2686996	1.2272420,-77.2679214	1.2286370,-77.2718277	1.2149820,-77.2789841	1.2238100,-77.2675400
1.1961250,-77.2723846	1.1970789,-77.2726440	1.2151430,-77.2744293	1.2021420,-77.2844543	1.2108090,-77.2826843	1.2096080,-77.2795563
1.1960930,-77.2703476	1.1971220,-77.2736740	1.2079560,-77.2735214	1.1998470,-77.2854767	1.2099510,-77.2854385	1.2102940,-77.2798996
1.1947620,-77.2691650	1.1968859,-77.2735214	1.1963930,-77.2826843	1.1962210,-77.2853928	1.2042450,-77.2856064	1.2078060,-77.2795944
1.1924450,-77.2692490	1.1957490,-77.2742920	1.2000180,-77.2811203	1.1881980,-77.2735062	1.2057030,-77.2842484	1.2057461,-77.2744904
1.1917160,-77.2701569	1.1892700,-77.2738113	1.1991820,-77.2824936	1.2048450,-77.2804565	1.2110020,-77.2800903	1.2053601,-77.2729034
1.1912230,-77.2687149	1.1946549,-77.2764587	1.2005980,-77.2825165	1.2043090,-77.2804718	1.2331949,-77.2707443	1.2041910,-77.2733154
1.1909220,-77.2678146	1.1976260,-77.2771149	1.2003620,-77.2833557	1.2080630,-77.2818451		1.2037190,-77.2765808
1.1797450,-77.2546692			1.2265770,-77.2678757		1.2055960,-77.2773438
1.1949550,-77.2689285					1.2077200,-77.2774963
1.1964570,-77.2682648					1.2105730,-77.2751846
1.1959640,-77.2690811					
1.2190690,-77.2695923					

Tabla 9-33. Coordenadas en latitud y longitud para las rutas de la 7 a la 12.

Ruta 7	Ruta 8	Ruta 9	Ruta 10	Ruta 11	Ruta 12
1.24788,-77.265499	1.24788,-77.265499	1.24788,-77.265499	1.24788,-77.265499	1.24788,-77.265499	1.24788,-77.265499
1.2037570,-77.2926331	1.2344069,-77.2653122	1.2086639,-77.2881775	1.2182430,-77.2852783	1.2248820,-77.2737808	1.2229730,-77.2679214
1.1970470,-77.2930069	1.2153890,-77.2804108	1.2061750,-77.2929459	1.2172880,-77.2836304	1.2201200,-77.2804108	1.2119679,-77.2989960
1.1981950,-77.2912674	1.2045020,-77.2931519	1.2103800,-77.2929916	1.2162369,-77.2790833	1.2191761,-77.2816772	1.2158720,-77.2921295
1.2018420,-77.2877121	1.2088780,-77.2923889	1.2124610,-77.2921982	1.2269419,-77.2675476	1.2212780,-77.2854385	1.2183610,-77.2909851
1.2052310,-77.2887421	1.2127510,-77.2865677	1.2138560,-77.2997284		1.2104660,-77.2993393	1.2243030,-77.2916565
1.2306319,-77.2677994	1.2140590,-77.2851715	1.2133840,-77.3011398		1.2174380,-77.2922363	1.2245390,-77.2914047
	1.2155720,-77.2808456	1.2147141,-77.2967606		1.2188330,-77.2889938	1.2232521,-77.2912674
	1.2157000,-77.2800217				1.2268130,-77.2964172
	1.2321759,-77.2709808				

Tabla 9-34. Coordenadas en latitud y longitud para las rutas de la 12 a la 18.

Ruta 13	Ruta 14	Ruta 15	Ruta 16	Ruta 17	Ruta 18
1.24788,-77.265499	1.24788,-77.265499	1.24788,-77.265499	1.24788,-77.265499	1.24788,-77.265499	1.24788,-77.265499
1.2214930,-77.2824020	1.1915020,-77.2637405	1.2303740,-77.2866745	1.2192620,-77.2662048	1.2237670,-77.2701187	1.2142950,-77.2643661
1.2254510,-77.2884750	1.1895980,-77.2625580	1.2309750,-77.2932968	1.2062830,-77.2645340	1.2071190,-77.2710342	1.2120529,-77.2631226
1.2257620,-77.2900620	1.1929870,-77.2619781	1.2243460,-77.2903366	1.2054030,-77.2635880	1.2046520,-77.2674255	1.2091140,-77.2597046
1.2269419,-77.2944870	1.1926121,-77.2613525	1.2229940,-77.2869797	1.2031291,-77.2634888	1.2048020,-77.2684860	1.2103800,-77.2591248
1.2323910,-77.2932053	1.1990100,-77.2610168	1.2237670,-77.2831192	1.1930680,-77.2630768	1.2030970,-77.2684326	1.2051030,-77.2612305
1.2305030,-77.2929840	1.2016270,-77.2544479	1.2268770,-77.2833023	1.1912870,-77.2653046	1.1980020,-77.2645187	1.2070440,-77.2621613
	1.1887560,-77.2389603	1.2271560,-77.2817841	1.1930031,-77.2651749	1.1946770,-77.2639313	1.2189610,-77.2535095
	1.2048450,-77.2254791		1.1945050,-77.2655716	1.1935830,-77.2725372	
	1.2126970,-77.2174988		1.1980129,-77.2687378	1.1968000,-77.2703705	
	1.2044590,-77.2101288		1.2055210,-77.2696152	1.2006910,-77.2698975	
	1.2122040,-77.2723007		1.2069480,-77.2715302	1.2010700,-77.2700424	
	1.2164840,-77.2709656			1.2102410,-77.2735519	
	1.2212780,-77.2687454				

9.9 CALCULO DE SOBRECOSTO DE LAS RUTAS DADAS LAS CONDICIONES TEMPORALES.

Según las rutas obtenidas con googlemaps, se realiza la tabla 9-35, identificando las calles de alta y media densidad usadas por cada ruta y los sectores a los cuales se dirige el vehículo de la ruta, para así asignar el sobrecosto en la matriz de distancias. (2,27 en calles consideradas de categoría o congestión alta y 1,95 en calles de congestión media).

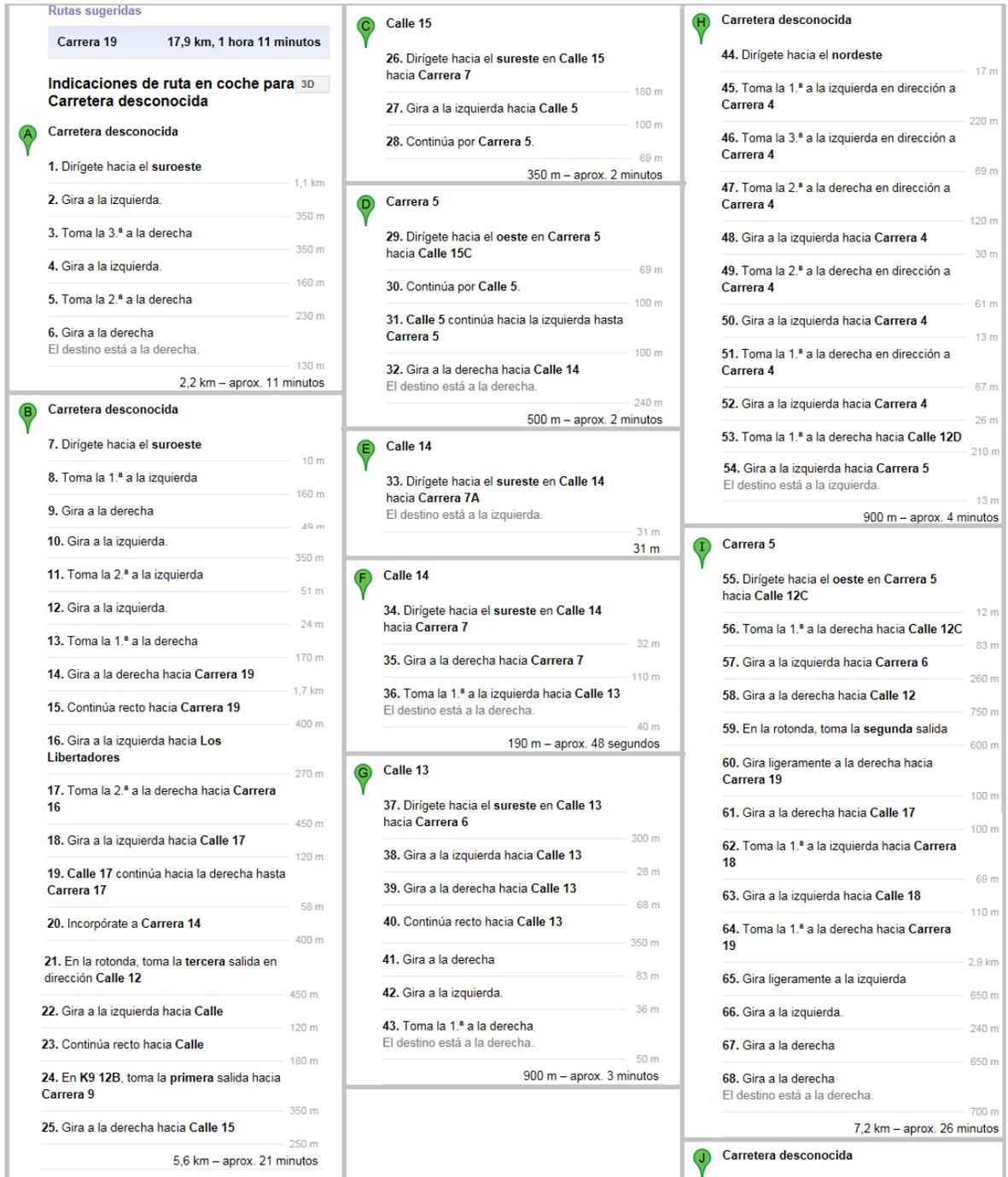


Figura 9-21. Ruta por calles de la ciudad de San Juan de Pasto para el vehículo 2 de Colácteos.

La figura 9-21, permite visualizar las diferentes calles usadas por las rutas del googlemaps y así poder definir el sobrecosto.

Tabla 9-35. Asignación de sobrecostos a las rutas

RUTA	Sector afectado (Inicio fin)	Correspondencia del sector de cada ruta en matriz de costos según coordenadas (LAT LONG) para los 154 sectores	Calles densas usadas por la ruta	Tipo de sobrecosto	Factor de sobrecosto
1	A B	1 a 96	Cr 19, Cr 14, Cr 13, Cll 20	Media, media, media, media	7,88
	B C	96 a 15	Cr 12, Cll 16, Cll 15	Alta, alta, media	6,49
	D E	104 a 66	Cll 15	Media	1,95
	H I	75 a 69	Cll 15	Media	1,95
	I J	69 a 74	Cr 19	Alta	2,27
	J K	74 a 64	Cll 15	Media	1,95
	K L	64 a 63	Cll 15	Media	1,95
	L M	63 a 65	Cll 15	Media	1,95
M N	65 a 143	Cll 15, Cr 19	Media, media	3,9	
2	B C	91 a 147	Cr 19, Cll 17, Cll 12, Cll 15	Media, Alta, alta, media	8,44
	C D	147 a 148	Cll 15	Media	1,95
	D E	148 a 17	Cll 15	Media	1,95
	H I	13 a 6	Cll 12	Alta	2,27
	I J	6 a 1	Cll 12, Cr 19, Cll 17, Cr 18	Alta, media, alta, media	8,44
3	B C	92 a 24	Cr 24, Av Santander	Alta, Alta	4,54
	C D	24 a 82	Av Santander, Cr 22, Cll 20	Alta, Alta, media	6,49
	D E	82 a 116	Cr 16, Cll 17, Cll 12	Media, Alta, Alta	6,49
	I J	7 a 1	Cr 19	Media	1,95
4	B C	102 a 2	Cr 19	Media	1,95
	F G	22 a 12	Cll 12	Alta	2,27
	G H	12 a 11	Av Boyacá	Alta	2,27
	H I	11 a 115	Av Boyacá	Alta	2,27
	I J	115 a 101	Cr 22, Cll 17, Cr 19, Cll 18	Alta, Alta, Media, Alta	8,76
5	A B	1 a 121	Cr 24, Cll 18	Alta, Alta	4,54
	B C	121 a 30	Cr 24, Av Boyacá	Alta, Alta	4,54
	C D	30 a 28	Cr 24	Alta	2,27
	D E	28 a 110	Cr 22	Alta	2,27
	E F	110 a 119	Cr 22	Alta	2,27
	F G	119 a 133	Av Boyacá, Cll 15	Alta, Media	4,22
	G H	133 a 103	Cll 16, Cr 22	Alta, Alta	4,54

6	B C	93 a 29	Cr 24, Cr 22	Alta, Alta	4,54
	C D	29 a 122	Cr 22, Clle 15	Alta, Media	4,22
	D E	122 a 10	Cr 22, Cr 20	Media, Media	3,9
	E F	10 a 18	Cr 19, Clle 17	Media, Alta	4,22
	F G	18 a 19	Clle 18, Clle 19	Alta, Alta	4,54
	G H	19 a 97	Clle 19, Clle 20	Alta, Media	4,22
	I J	99 a 9	Clle 16, Clle 12	Alta, Alta	4,54
	J K	9 a 149	Clle 16	Alta	2,27
	K L	149 a 154	Clle 16, Clle 17, Cr 19, Clle 18, Clle 20	Alta, Alta, Media, Alta, Media	10,71
	K M	154 a 1	Clle 19	Alta	2,27
7	E F	109 a 107	Cr 22	Alta	2,27
8	B C	124 a 31	Clle 16, Cr 24	Alta, Alta	4,54
	C D	31 a 49	Clle 17, Cr 24	Alta, Alta	4,54
	F G	53 a 60	Clle 12	Alta	2,27
	G H	60 a 137	Cr 27, Clle 16	Media, Alta	4,22
	H I	137 a 136	Clle 17, Cr 27	Alta, Media	4,22
	I J	136 a 125	Clle 17, Cr 26	Alta, Media	4,22
	9	A B	1 a 150	Cr 27	Media
B C		150 a 108	Av Panamericana	Media	5,85
H I		55 a 1	Clle 15, Cr 26	Media, Media	3,9
10	A B	1 a 26	Cr 27, Clle 16	Media, Alta	4,22
	B C	26 a 27	Clle 15	Media	1,95
	C D	27 a 135	Clle 15, Cr 26, Clle 18	Media, Media, Alta	6,17
	D E	135 a 140	Cr 27, Clle 19	Media, Alta	4,22
11	B C	120 a 25	Clle 18	Alta	2,27
	C D	25 a 134	Clle 19	Alta	2,27
	D E	134 a 61	Clle 16	Alta	2,27
	E F	61 a 57	Clle 16	Alta	2,27
	H I	151 a 1	Clle 15, Cr 26	Media, Media	3,9
12	B C	100 a 58	Cr 26, Cr 30, Clle 18	Media, Media, Alta	6,17
	E F	153 a 89	Clle 12	Alta	2,27
	H I	90 a 132	Clle 15, Clle 16	Media, Alta	4,22
	I J	132 a 1	Clle 16, Clle 18, Clle 17, Cr 26	Alta, Alta, Alta, Media	8,76
13	A B	1 a 51	Clle 18	Alta	2,27
	B C	51 a 86	Clle 17	Alta	2,27
	C D	86 a 88	Clle 17, Clle 16	Alta, Alta	4,54
	D E	88 a 130	Clle 16	Alta	2,27
	E F	130 a 127	Clle 18	Alta	2,27
	F G	127 a 128	Clle 18	Alta	2,27
	G H	128 a 1	Clle 17, Cr 26	Alta, Media	4,22
14	A B	1 a 72	Cr 19, Clle 18, Clle	Media, Alta,	6,49

			17	Alta	
	C D	71 a 78	Clle 16	Alta	2,27
	D E	78 a 79	Clle 16	Alta	2,27
	E F	79 a 32	Clle 16, Clle 19	Alta, Alta	4,54
	F G	32 a 33	Clle 19, Clle 20	Alta, Media	4,22
	K L	42 a 106	Cr 19	Media	1,95
	M N	141 a 139	Cr 24	Alta	2,27
15	A B	1 a 85	Clle 19	Alta	2,27
	B C	85 a 131	Clle 18	Alta	2,27
	C D	131 a 87	Clle 18, Clle 16	Alta, Alta	4,54
	D E	87 a 52	Clle 17, Cr 40	Alta, Media	4,22
	E F	52 a 126	Clle 17, Clle 18, Clle 16	Alta, Alta, Alta	6,81
	G H	84 a 129	Clle 19	Alta	2,27
	H I	129 a 1	Clle 19, Cr 26	Alta, Media	4,22
16	E F	34 a 73	Clle 19, Clle 16	Alta, Alta	4,54
	I J	62 a 77	Clle 17, Clle 18	Alta, Alta	4,54
	J K	77 a 80	Clle 17, Clle 18	Alta, Alta	4,54
	K L	80 a 81	Clle 18, Clle 20	Alta, Media	4,22
	L M	81 a 1	Clle 20, Cr 19	Media, Alta	4,22
17	B C	138 a 83	Cr 19	Media	1,95
	E F	45 a 76	Clle 20	Media	1,95
	F G	76 a 46	Clle 19, Clle 18	Media	1,95
	G H	46 a 50	Clle 18	Media, Alta	4,22
	H I	50 a 20	Clle 17	Alta	2,27
	K L	98 a 8	Clle 18	Alta	2,27
	L M	8 a 142	Cr 19	Media	1,95
	M N	142 a 1	Cr 19	Media	1,95
18			Villa nueva, nuevo sol.	Baja	

Con base en la tabla 9-35 se establece el costo de las rutas con las medidas reales de distancia entre sectores más el factor de sobrecosto por congestión vehicular.

Tabla 9-36. Costos de cada ruta incluyendo la congestión vehicular en horas pico

RUTA	Correspondencia del sector de cada ruta en matriz de costos según coordenadas (LAT LONG) para los 153 sectores y el depósito	Factor de sobrecosto	Costo en distancia entre sectores Km	Distancia afectada por sobrecosto para los sectores congestionados Km.
1	1 a 96	7,88	8,863	69,84044

	96 a 15	6,49	1,179	7,65171
	104 a 66	1,95	0,438	0,8541
	75 a 69	1,95	0,104	0,2028
	69 a 74	2,27	1,772	4,02244
	74 a 64	1,95	2,322	4,5279
	64 a 63	1,95	0,282	0,5499
	63 a 65	1,95	0,139	0,27105
	65 a 143	3,9	3,324	12,9636
2	91 a 147	8,44	3,831	32,33364
	147 a 148	1,95	0,163	0,31785
	148 a 17	1,95	0,031	0,06045
	13 a 6	2,27	0,51	1,1577
	6 a 1	8,44	12,978	109,53432
3	92 a 24	4,54	1,341	6,08814
	24 a 82	6,49	0,994	6,45106
	82 a 116	6,49	2,189	14,20661
	7 a 1	1,95	12,51	24,3945
4	102 a 2	1,95	4,286	8,3577
	22 a 12	2,27	1,88	4,2676
	12 a 11	2,27	0,071	0,16117
	11 a 115	2,27	0,7	1,589
	115 a 101	8,76	2,693	23,59068
5	1 a 121	4,54	10,329	46,89366
	121 a 30	4,54	0,777	3,52758
	30 a 28	2,27	0,37	0,8399
	28 a 110	2,27	0,799	1,81373
	110 a 119	2,27	0,504	1,14408
	119 a 133	4,22	0,903	3,81066
	133 a 103	4,54	2,177	9,88358
6	93 a 29	4,54	1,707	7,74978
	29 a 122	4,22	0,404	1,70488
	122 a 10	3,9	0,542	2,1138
	10 a 18	4,22	0,85	3,587
	18 a 19	4,54	0,379	1,72066
	19 a 97	4,22	0,192	0,81024
	99 a 9	4,54	0,92	4,1768
	9 a 149	2,27	0,272	0,61744
	149 a 154	10,71	0,705	7,55055
	154 a 1	2,27	8,14	18,4778
7	109 a 107	2,27	0,499	1,13273
8	124 a 31	4,54	1,613	7,32302
	31 a 49	4,54	2,107	9,56578
	53 a 60	2,27	0,353	0,80131
	60 a 137	4,22	0,524	2,21128
	137 a 136	4,22	0,606	2,55732
	136 a 125	4,22	1,496	6,31312

9	1 a 150	1,95	11,278	21,9921
	150 a 108	5,85	0,688	4,0248
	55 a 1	3,9	10,962	42,7518
10	1 a 26	4,22	9,88	41,6936
	26 a 27	1,95	0,574	1,1193
	27 a 135	6,17	0,72	4,4424
	135 a 140	4,22	1,876	7,91672
11	120 a 25	2,27	1,916	4,34932
	25 a 134	2,27	0,315	0,71505
	134 a 61	2,27	0,538	1,22126
	61 a 57	2,27	2,443	5,54561
	151 a 1	3,9	9,754	38,0406
12	100 a 58	6,17	4,11	25,3587
	153 a 89	2,27	0,928	2,10656
	90 a 132	4,22	0,548	2,31256
	132 a 1	8,76	10,279	90,04404
13	1 a 51	2,27	9,539	21,65353
	51 a 86	2,27	1,089	2,47203
	86 a 88	4,54	0,29	1,3166
	88 a 130	2,27	0,445	1,01015
	130 a 127	2,27	1,126	2,55602
	127 a 128	2,27	0,407	0,92389
	128 a 1	4,22	9,118	38,47796
14	1 a 72	6,49	10,839	70,34511
	71 a 78	2,27	0,74	1,6798
	78 a 79	2,27	0,353	0,80131
	79 a 32	4,54	1,048	4,75792
	32 a 33	4,22	1,024	4,32128
	42 a 106	1,95	8,883	17,32185
	141 a 139	2,27	0,423	0,96021
15	1 a 85	2,27	7,934	18,01018
	85 a 131	2,27	0,925	2,09975
	131 a 87	4,54	1,208	5,48432
	87 a 52	4,22	0,572	2,41384
	52 a 126	6,81	0,831	5,65911
	84 a 129	2,27	1,48	3,3596
	129 a 1	4,22	9,314	39,30508
16	34 a 73	4,54	1,43	6,4922
	62 a 77	4,54	1,075	4,8805
	77 a 80	4,54	1,409	6,39686
	80 a 81	4,22	0,257	1,08454
	81 a 1	4,22	8,354	35,25388
17	138 a 83	1,95	2,373	4,62735
	45 a 76	1,95	0,3	0,585
	76 a 46	1,95	0,895	1,74525
	46 a 50	4,22	0,557	2,35054

	50 a 20	2,27	1,555	3,52985
	98 a 8	2,27	0,031	0,07037
	8 a 142	1,95	1,38	2,691
	142 a 1	1,95	7,907	15,41865
18		0	0	0

Con base en el costo calculado en las calles de congestión alta y media se pasa a calcular el costo total de las rutas, esto se realiza con el costo obtenido en cada ruta según googlemaps, más el costo afectado por el factor de sobrecosto menos el costo de las calles afectadas puesto que este ya se incluye en la distancia afectada por el sobrecosto (Columna 3 de la tabla 9-36).

Tabla 9-37. Costo de cada ruta y total afectado por sobrecosto de tráfico en hora pico

RUTA	Distancia de cada ruta en Km	Distancia Km afectada por factor de sobrecosto
1	20,6	103,06094
2	17,3	143,19096
3	17,1	51,20631
4	20,2	48,53615
5	14,3	66,35419
6	17,8	52,19795
7	15,5	16,13373
8	16,9	38,97283
9	17	62,8407
10	11,7	53,82202
11	15,8	50,70584
12	19,2	123,15686
13	16,2	62,59618
14	35,1	111,97748
15	18,6	72,66788
16	20	61,58298
17	18,5	34,52001
18	16,3	16,3
SUMA	328,1	1169,82
RELACIÓN		3,565