



Escuela Superior Politécnica del Litoral
Instituto de Ciencias Matemáticas
Ingeniería en Logística y Transporte

“Implementación de un Problema de Ruteo Vehicular con Ventanas de Tiempo (VRPTW) en una empresa de venta de agroquímicos, sucursal Milagro.”

**Informe del Proyecto de Graduación
(dentro de una materia de la malla)**

Previo a la obtención del Título de:
Ingeniero en Logística y Transporte

Presentado por:
Cristofer Morán Villa
Jorge Núñez Ginez

Guayaquil – Ecuador

Año
2012

Agradecimiento

Agradecemos en primer lugar a Dios, quien nos otorgó la vida, y la capacidad de poder aprender y desarrollarnos como profesionales, y con sus bendiciones ha llenado nuestras vidas de triunfos y éxito.

Agradecemos a nuestros padres, quienes con su paciencia y amor, nos guiaron siempre por el camino del bien en la vida, siendo nuestros amigos y consejeros. De la misma forma queremos agradecer de manera especial a los profesores que mediante sus conocimientos y consejos ayudaron de manera incondicional en nuestra formación como profesionales, así como también al Mat. Fernando Sandoya y al Msc. Fabricio Echeverría, quienes fueron pilares fundamentales de conocimientos para el desarrollo de esta tesis.

Dedicatoria

Dedico este trabajo a mis padres quienes gracias a su apoyo y comprensión pude culminar mis estudios universitarios, a mi esposa, Lissette e hija, Llean, por ser el motivo y las fuerzas de no dejarme vencer en los momentos más difíciles de mi vida, y con su amor me supieron guiar hacia mis metas profesionales.

Cristofer Morán Villa.

Dedico este trabajo a mis padres que siempre me brindaron su apoyo incondicional, en los momentos más difíciles de mi vida y supieron guiarme y comprenderme con su paciencia y amor; y al mismo tiempo a mi novia Gabriela, quien siempre estuvo a mi lado en mi formación profesional.

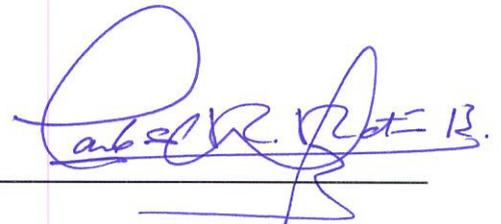
Jorge Núñez Ginez.

Tribunal de Graduación



M.Sc. Fabricio Echeverría Briones

DIRECTOR DEL PROYETO DE
GRADUACIÓN
(dentro de una materia de la malla)

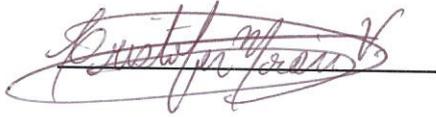


M.Sc. Carlos Martín Barreiro

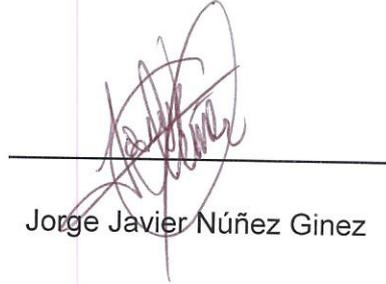
DELEGADO DEL ICM

Declaración Expresa

“La responsabilidad del contenido de este trabajo final de graduación, nos corresponde exclusivamente; y el patrimonio intelectual del mismo a la Escuela Superior Politécnica del Litoral”



Cristofer David Morán Villa



Jorge Javier Núñez Ginez

Resumen

Hoy en día las empresas de bienes y/o servicios abordan problemas dentro de su cadena logística, existiendo la importancia de implementar sistemas que ayuden a la mejora continua de esta gestión y permita a las empresas ser más competitivas satisfaciendo las necesidades de sus clientes. Por esta razón la presente investigación ha sido desarrollada para optimizar los tiempos de distribución de productos de una empresa de venta de agroquímicos. Esta empresa tiene el firme deseo de mejorar su gestión logística. Para poder cumplir con este objetivo, se plantea un conjunto de rutas entre una serie de clientes y/o destinos, de las cuales deberá ser elegida la mejor; es decir, la que permita realizar la distribución de los productos agroquímicos en el menor tiempo posible y a su vez minimizar sus costos de distribución. A este tipo de problema se lo conoce como el Problema de Ruteo Vehicular con Ventanas de Tiempo (VRPTW por sus siglas en inglés).

En el primer capítulo daremos a conocer los problemas que se presentan hoy en día en las empresas que realizan distribución primaria y secundaria a sus clientes, junto con la importancia que se da en los altos niveles gerenciales, tratando de llegar a la mejora continua en la etapa de la entrega de pedidos. Además, dentro de este capítulo se detallará los parámetros a considerar y las variables involucradas del problema planteado.

Dentro del segundo capítulo se muestra las definiciones y formulaciones matemáticas del problema abordado (VRPTW) y los algoritmos utilizados entre éstos el algoritmo del Vecino más Cercano y del Recocido Simulado que es algoritmo de mejora seleccionado para el desarrollo de este proyecto; así como también, se detallan ejemplos que demuestran la eficiencia de los mismos.

Por último en los siguientes capítulos, se presenta el detalle de los diferentes puntos abordados para la elaboración del problema, como el software de programación, funciones utilizadas en la ejecución del programa principal obteniendo resultados positivos en comparación al sistema actual de la empresa los mismos que fueron analizados presentando así, las conclusiones y recomendaciones respectivas de éste problema en particular.

Índice General

Agradecimiento.....	II
Dedicatoria	III
Tribunal de Graduación.....	IV
Declaración Expresa.....	V
Resumen	VI
Índice General.....	VII
Índice de Ilustraciones.....	IX
Índice de Tablas	X
Abreviaturas	XI
1. Antecedentes.....	- 1 -
1.1 Introducción	- 1 -
1.2 Descripción del problema.....	- 2 -
1.2.1 Situación Actual.....	- 2 -
1.2.2 Antecedentes de Teoría	- 7 -
1.2.3 Parámetros a Considerar	- 9 -
1.2.4 Variables Involucradas	- 9 -
1.3 Objetivos	- 9 -
2. Marco Teórico.....	- 10 -
2.1 Introducción	- 10 -
2.2 Definiciones Matemáticas y Ejemplos.....	- 10 -
2.2.1 Formulación Matemática del TSP.....	- 10 -
2.2.1.1 Comparación entre Métodos de Resolución de Problemas.....	- 11 -
2.2.1.1.1 Ejemplo No. 1.....	- 12 -
2.2.1.1.1.1 Aplicación de Algoritmo del Vecino más Cercano para Ejemplo 1 .	- 14 -
2.2.1.2 Ejemplo No. 2.....	- 15 -
2.2.1.2.1 Aplicación del Algoritmo del Vecino más Cercano para el Ejemplo 2.	- 18 -
2.2.2 Definición Matemática del VRPTW	- 19 -
2.3.1 Modelo Matemático del VRPTW.....	- 21 -
2.3.2 Complejidad Computacional del VRPTW.....	- 23 -

2.4 Algoritmo de mejoramiento Recocido Simulado (Simulated Annealing)....	- 23 -
2.4.1 Iteración básica.....	- 24 -
2.4.2 Vecindario de un estado	- 24 -
2.4.3 Probabilidad de Transición	- 24 -
2.4.4 Protocolo de recocido	- 25 -
3. Elaboración del Programa.....	- 26 -
3.1 Selección del Software de Programación.....	- 26 -
3.2 Estructura de los Datos	- 26 -
3.3 Funciones	- 28 -
3.3.1 Función Crear Coordenada de Cliente	- 28 -
3.3.2 Función Grafo Cliente.....	- 28 -
3.3.3 Función Crear Matriz de Distancias	- 28 -
3.3.4 Función Vecino Más Cercano	- 29 -
3.3.5 Función Crear Ruta.....	- 29 -
3.3.6 Función Crear Rutas Iniciales	- 29 -
3.3.7 Función Graficar Solución.....	- 29 -
3.3.8 Función Calcular Costo Ruta.....	- 30 -
3.3.9 Función Calcular Costo Ruteo	- 30 -
3.3.10 Función Scheduling Ruta.....	- 30 -
3.3.11 Función Calcular Penalización.....	- 30 -
3.3.12 Función Costo Total Ruta	- 30 -
3.3.13 Función Costo Total Ruteo.....	- 31 -
3.3.14 Función Generar Vecino	- 31 -
3.5 Programa Principal.....	- 32 -
4. Desarrollo y Análisis del Problema	- 34 -
4.1 Tablas de Información	- 34 -
5. Resolución del Problema.....	- 36 -
6. Conclusiones y Recomendaciones.....	- 42 -
7. Referencias Bibliográficas.....	- 44 -

Índice de Ilustraciones

Ilustración 1: Flujo de Distribución	- 2 -
Ilustración 2: Ubicación geográfica de los nodos principales de los clientes.	- 5 -
Ilustración 3: Esquema de Trabajo de Empresa.....	- 6 -
Ilustración 4: Grafo Inicial de Ejemplo No. 1	- 12 -
Ilustración 5: Grafo Solución del Ejemplo No. 1	- 13 -
Ilustración 6: Grafo Inicial de Vecino más Cercano Ejemplo No. 1.....	- 14 -
Ilustración 7: Pasos del Algoritmo del Vecino más Cercano del Ejemplo No. 1....	- 15 -
Ilustración 8: Grafo Solución Algoritmo del Vecino más Cercano Ejemplo No. 1..	- 15 -
Ilustración 9: Grafo Inicial de Ejemplo No. 2	- 16 -
Ilustración 10: Grafo Solución del Ejemplo No. 2.....	- 17 -
Ilustración 11: Grafo Inicial de Vecino más Cercano Ejemplo No. 1.....	- 18 -
Ilustración 12: Pasos del Algoritmo del Vecino más Cercano del Ejemplo No. 2..	- 18 -
Ilustración 13: Función de Distribución Actual de Empresa.....	- 34 -
Ilustración 14: Ruteo Empírico	- 36 -
Ilustración 15: Ruta 1	- 37 -
Ilustración 16: Ruta 2.....	- 38 -
Ilustración 17: Ruta 3.....	- 38 -
Ilustración 18: Ruta 4.....	- 38 -
Ilustración 19: Ruteo con el Algoritmo de recocido Simulado.....	- 39 -

Índice de Tablas

Tabla 1: Distancias entre Clientes de Ejemplo No. 1	- 12 -
Tabla 2: Distancias entre clientes de Ejemplo No. 2	- 16 -
Tabla 3: Estructura de ingreso de datos	- 27 -
Tabla 4: Tabla de Distancias de Clientes	- 35 -
Tabla 5: Scheduling Ruta 1	- 40 -
Tabla 6: Scheduling Ruta 2.....	- 41 -
Tabla 7: Scheduling Ruta 3.....	- 41 -
Tabla 8: Scheduling Ruta 4.....	- 41 -

Abreviaturas

VRP	Problema de Enrutamiento Vehicular (Vehicle Routing Problem)
VRPTW	Problema de Enrutamiento Vehicular con Ventanas de Tiempo (Vehicle Routing Problem with Time Windows)
TSP	Problema del Agente Viajero (Travel Salesman Problem)
SA	Recocido Simulado (Simulated Annealing)
NP	Tiempo Polinomial No Determinista (Nondeterministic Polynomial Time)
CVRP	Problema del Enrutamiento Vehicular Capacitado (Capacitated Vehicle Routing Problem)
PLE	Programación Lineal Entera (Integer Linear Programming)

1. Antecedentes

1.1 Introducción

En la mayoría de las empresas, siempre se tienen costos elevados y pérdidas, esto se debe a la mala gestión logística. Para una empresa de distribución uno de los problemas principales es la inadecuada entrega de mercaderías, que por lo general se basan en sistemas rústicos que trabajan de manera empírica debido a la falta de conocimiento de la aplicación de los modelos matemáticos que ayudan a resolver problemas de ruteo, los cuales buscan optimizar los recursos para así minimizar costos y maximizar utilidad, que es siempre el objetivo de las empresas.

Uno de los modelos más conocidos, debido a su gran aproximación a lo que es una solución óptima, es el VRP (Vehicle Routing Problem), el cual consiste en crear rutas al costo más bajo posible.

El objetivo del VRP es determinar un conjunto de rutas óptimas para que una flota de vehículos localizados en uno o varios depósitos atiendan la demanda de bienes o servicios a un conjunto clientes distribuidos geográficamente en un periodo dado, utilizando para ello una red vial disponible.

En este proyecto se han aplicado las distintas técnicas utilizadas de la investigación de operaciones para poder determinar la solución a un problema de planificación de rutas de una empresa dedicada a la venta de agroquímicos. Las técnicas aplicadas permiten determinar a qué cliente vamos a visitar, el cual va a estar relacionado a una ruta para cada sector establecido por el programa.

Este problema que se ha abordado ha sido considerado en dos etapas para su resolución; la primera, fijar el centro de distribución como nodo principal y la segunda, implementar el ruteo vehicular para cada sector previamente establecido, asociando una ventana horaria que deberá respetar cada vehículo para el sector y ruta asignada; teniendo en cuenta el tiempo de salida del camión de su centro de distribución, capacidad, distancias, horas de inactividad y hora de llegada.

1.2 Descripción del problema

1.2.1 Situación Actual

Actualmente, la empresa maneja dos tipos de transporte, el primero es el transporte primario que consiste en el envío de agroquímicos desde el centro de distribución ubicado en la ciudad de Guayaquil a cada una de las sucursales; y el transporte secundario, que es el motivo de estudio de este trabajo de graduación, el cual radica en la entrega de los productos desde las sucursales hasta los clientes finales, que son los minoristas, tal como se detalla en la ilustración 1 que representa el flujo de distribución antes descrito.

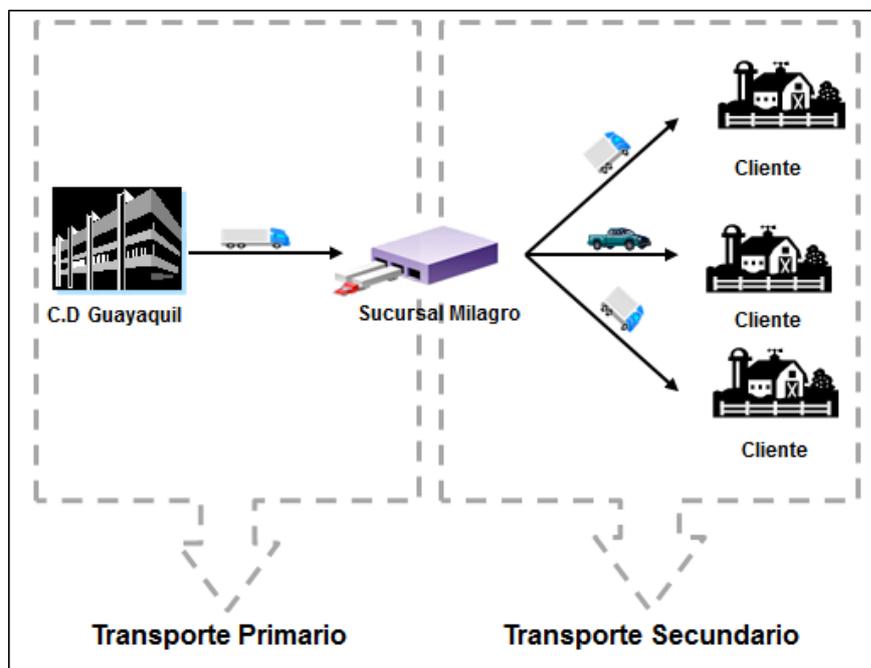


Ilustración 1: Flujo de Distribución

La sucursal seleccionada para este proyecto se encuentra situada en la ciudad de Milagro, en la actualidad realizan el servicio de distribución con dos camiones de 5 toneladas y una camioneta de 2.5 toneladas, que se encargan de la entrega de los productos a los diferentes consumidores existentes en la zona y sus alrededores.

La empresa utiliza la camioneta para entregas pequeñas y de emergencia, que justifiquen su despacho.

La empresa se maneja con una lista de clientes pertenecientes a la industria del agro. La empresa cuenta con clientes que manejan fincas y/o haciendas, cuyas facturas comerciales generalmente son solicitadas en sus oficinas ubicadas en la ciudad de Guayaquil y el producto es requerido para su entrega directa en la hacienda y/o almacén.

Al momento la sucursal ubicada en la ciudad de Milagro cuenta con 35 clientes, los cuales por motivos de confidencialidad de la empresa no se detalla la dirección exacta ni el nombre de la hacienda; sin embargo, en la programación realizada si se considera esta información para el cálculo de las distancias, a continuación el detalle de los recintos y ciudades a los cuales se les ha asignado una letra que va desde A hasta J para efectos de gráfica, encontrándose dentro de estos nodos principales sus respectivos números de clientes y dirección general de los mismos:

- **Milagro (A y B):**
 - Cliente 1, *Depósito* (Km 1 Vía Km 26)
 - Cliente 2 (Km 4 ½ Vía Km 26)
 - Cliente 3 (Vía Km26)
 - Cliente 4 (Vía Km 26)
 - Cliente 5 (Banco de Arena)
- **El Deseo (C):**
 - Cliente 6 (Vía Cone)
 - Cliente 7 (Vía Vuelta Larga)
- **Kilómetro 26 (D):**
 - Cliente 8 (Taura)
 - Cliente 9 (Taura)
- **San Carlos (E):**
 - Cliente 10

- **La Troncal (F):**
 - Cliente 11 (Vía Puerto Inca-Troncal)
 - Cliente 12
 - Cliente 13
 - Cliente 14
 - Cliente 15
 - Cliente 16
- **El Triunfo (G):**
 - Cliente 17 (El Triunfo)
 - Cliente 18 (Vía Km 26-El Triunfo)
 - Cliente 19 (Km 26 Vía a El Triunfo)
 - Cliente 20 (Km 32 Vía a El Triunfo)
 - Cliente 21 (Vía Triunfo-Pedrero)
 - Cliente 22 (Vía Triunfo-Pedrero)
 - Cliente 23 (Vía Triunfo-San Pedro)
- **Bucay (H):**
 - Cliente 24 (Km 12, Vía El Triunfo-Bucay)
 - Cliente 25 (Km 4 Vía El Triunfo-Bucay)
 - Cliente 26 (Km 5 Vía El Triunfo-Bucay)
 - Cliente 27 (Km 8 Vía El Triunfo-Bucay)
 - Cliente 28 (Km 7 Vía El Triunfo-Bucay)
 - Cliente 29 (Km 20 Vía El Triunfo-Bucay)
 - Cliente 30 (Mathilde Esther)
 - Cliente 31 (El Rosario)
- **Mariscal Sucre (I):**
 - Cliente 32 (Vía Mariscal-Garaicoa)
- **Naranjito (J):**
 - Cliente 33
 - Cliente 34 (Vía Naranjito Bucay)
 - Cliente 35 (Vía Naranjito Bucay)
 - Cliente 36 (Vía Naranjito Bucay)

A continuación en la ilustración 2 se muestra el detalle de la ubicación geográfica de los nodos principales.

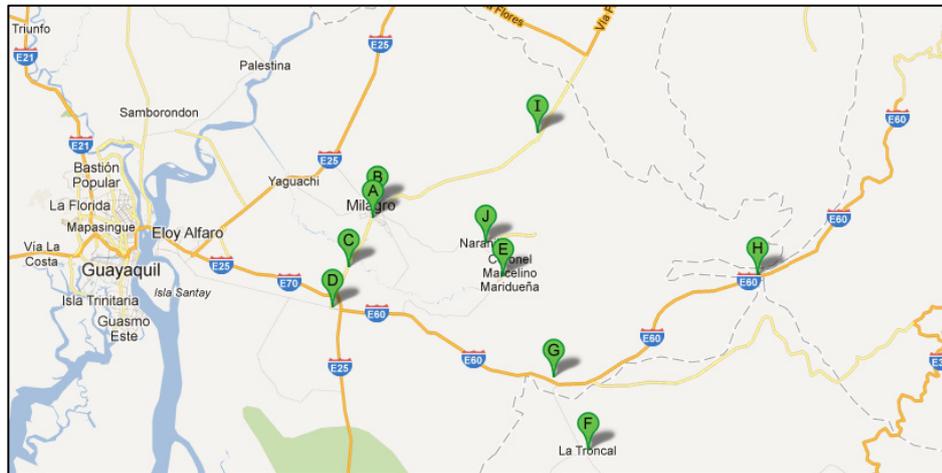


Ilustración 2: Ubicación geográfica de los nodos principales de los clientes.

El trabajo actual de la empresa funciona con la recepción de las llamadas de los clientes para realizar su pedido y facturarlo, luego al momento de la facturación se emite una guía de despacho a bodega donde se detallan los productos requeridos por el cliente, posteriormente en bodega se consolida la carga por cliente y tiempo de llamada, para así proceder a cargar el camión con el producto hasta su máxima capacidad o metraje.

Actualmente, el ruteo de la distribución se maneja de manera empírica; es decir, el chofer del camión, dependiendo de la prioridad de los clientes, es quien se encarga de entregar los productos. Además, hay que considerar que la empresa maneja una ruta conforme se vayan dando los pedidos, lo que significa que no existe una planificación adecuada con los pedidos que generan los clientes. A continuación en la ilustración 3 se muestra la rutina estándar de recepción de pedidos representada mediante un diagrama de flujo.

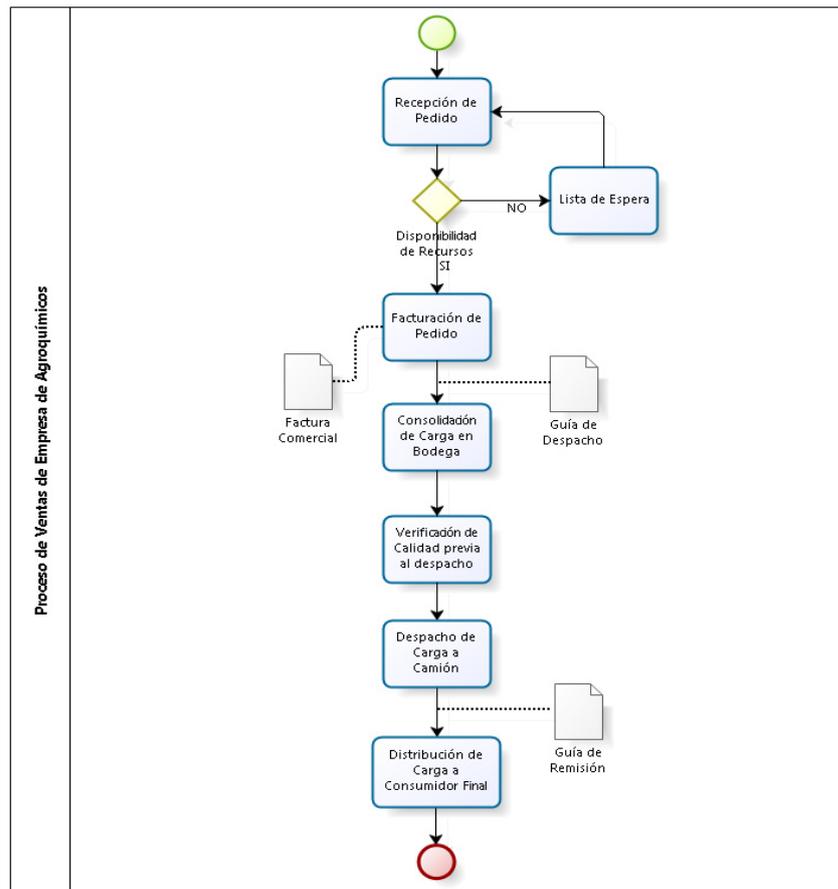


Ilustración 3: Esquema de Trabajo de Empresa.

En el presente trabajo se desarrollará una aplicación que permita a la empresa efectuar adecuadamente la planificación de sus rutas; es decir, que no se realice de manera empírica como actualmente lo hace. Para esto, se implementará un ruteo vehicular acorde a la situación actual de la empresa considerando factores tales como: demanda, capacidad de vehículos, ubicación geográfica y tiempos de servicio de sus clientes.

La mejora en la planificación de carga, permitirá a la empresa disminuir los costos asociados a la operación de distribución que actualmente se manejan de manera inapropiada; así mismo, al optimizar la distancia total recorrida de la red de distribución repercutirá directamente en los tiempos de entrega, lo cual permitirá visitar mayor cantidad de clientes mejorando el servicio brindado y por tanto ser más competitivos en el mercado.

1.2.2 Antecedentes de Teoría

Algoritmo, es un conjunto de pasos estructurados para realizar una tarea ya sea ésta simple o compleja.

Heurística, se refiere a la experiencia basada en técnicas de resolución de problemas, el aprendizaje y descubrimiento. Cuando una búsqueda exhaustiva no es práctica, los métodos heurísticos se utilizan para acelerar el proceso de encontrar una solución satisfactoria [7].

Metaheurística, son métodos de solución que orquestan una interacción entre los procedimientos de mejora local y estrategias de alto nivel para crear un proceso capaz de escapar de óptimos locales y realizar una búsqueda robusta en el espacio de soluciones factible del problema [3].

Problema del Agente Viajero “TSP” (Travelling Salesman Problem), es uno de los problemas más estudiados en optimización combinatoria, se dispone de un solo vehículo que debe visitar a todos los clientes en una sola ruta y a un costo mínimo. No suele haber un depósito y si lo hubiera no se distingue de los clientes, no hay demanda asociada a los clientes y tampoco hay restricciones temporales [6].

Heurística del Vecino Más Cercano, es un algoritmo que consiste en realizar el menor desplazamiento en distancias de un conjunto finito de puntos o individuos.

Recocido Simulado (Simulated Annealing), es una de las técnicas más flexibles disponibles para resolver problemas combinatorios. La principal ventaja es que puede ser aplicado a grandes problemas, independientemente de las condiciones de diferenciabilidad, continuidad y convexidad que normalmente se requiere en los métodos de optimización convencionales [5].

Ventanas de tiempo, es el período entre una hora de inicio y una hora de finalización en el que una ruta debería visitar una ubicación de la red de distribución, existen dos tipos de ventana de tiempo y estas son:

Duras: Especifican para cada cliente un intervalo de tiempo en el que forzosamente debe ser visitado (restricciones en el modelo).

Suaves: Especifican un intervalo de tiempo en el que un cliente debe ser visitado, asociado a una función de costo por retrasos/adelantos (penalizaciones en la función objetivo).

Para este proyecto se trabajó con ventanas de tiempo suaves considerando costos de penalización.

Vehicle Routing Problems with Time Windows (VRPTW), es un problema de ruteo de vehículos el cual tiene por objetivo la generación óptima de rutas de entrega a partir de un conjunto estático de pedidos, considerando que cada pedido tiene una locación y una banda horaria donde se debe realizar la entrega [2].

Por consiguiente, el VRPTW consiste en encontrar un conjunto de posibles rutas dentro de la región de soluciones factibles, donde cada cliente sea atendido por una sola ruta de su ventana de tiempo al mínimo costo de las rutas halladas; no obstante, este método no garantiza que la solución encontrada sea la óptima; sin embargo, esto servirá como solución inicial para el estudio de técnicas robustas como las Metaheurísticas, que si bien es cierto no consiguen la solución óptima pero si una aproximación bastante buena en un tiempo de cálculo moderado.

Existen medidas a considerar para la resolución del VRPTW que deben cumplirse como parte de la solución factible encontrada, las cuales se detallan a continuación:

- Vehículos con capacidad homogénea.
- Ventanas de tiempos para cada cliente.
- Los clientes pueden ser atendidos por más de un vehículo.
- Velocidad promedio para flota de vehículos.

- Subconjunto de caminos que pueden ser utilizados por más de un vehículo.
- Utilización de distancias basadas en la Métrica de Manhattan que fueron preestablecidas con la ayuda de GOOGLE-EARTH.
- Cada cliente debe ser atendido por una sola ruta dentro de su ventana de tiempo.

1.2.3 Parámetros a Considerar

- Visitas (nodos con demanda por parte del cliente).
- Depósitos (lugar donde comienzan y terminan las rutas).
- Localizaciones geográficas (tiempo / distancia) de un nodo a otro.
- Tiempos.

1.2.4 Variables Involucradas

- Demanda y localización de los clientes.
- Ubicación de los centros de distribución.
- Capacidad y tipos de vehículos disponibles.
- Tiempo de transportación.
- Descripción de las vías de transporte.

1.3 Objetivos

Los objetivos que se desean alcanzar mediante la elaboración de este proyecto son:

- Disminuir los tiempos de entrega y distancias recorridas, por parte de la empresa hacia los consumidores.
- Maximizar el nivel de servicio a los clientes, optimizando la utilización de los recursos.
- Buscar la aplicación del objetivo general de la logística: maximización de nivel del servicio y minimización de costos.
- Elaborar rutas de acuerdo al orden de los pedidos por parte de los consumidores.
- Balancear rutas, para el tiempo de viaje y carga del vehículo.
- Minimizar la penalización asociada con un servicio parcial de los clientes.

2. Marco Teórico

2.1 Introducción

El problema considerado es el ruteo vehicular de una empresa dedicada a la venta de agroquímicos cuya sucursal está ubicada en la ciudad de Milagro. En ese sentido, se generaron rutas con ventanas de tiempos para atender la demanda de los clientes con el fin de minimizar los costos totales de transportación, los cuales dependen de la distancia total recorrida o tiempo de viaje.

Para cumplir con los objetivos del problema se implementó la heurística del vecino más cercano, con un algoritmo de mejora, conocido como Recocido Simulado (Simulated Annealing, S.A.).

Conforme se vayan dando los pedidos por parte de los clientes, se genera el orden de secuenciación de la entrega de los productos; por ser ésta la política con la que trabaja la empresa actualmente.

2.2 Definiciones Matemáticas y Ejemplos

En general, es muy difícil aplicar técnicas exactas para encontrar la solución óptima, es por eso que existen heurísticas para determinar buenas soluciones.

2.2.1 Formulación Matemática del TSP

El Problema del Agente Viajero (*Travelling Salesman Problem*) se lo formula de la siguiente manera [8]:

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$
$$s. a. \sum_{j=1}^n x_{ij} = 1; \quad i = 1, 2, \dots, n$$

$$\sum_{i=1}^n x_{ij} = 1; \quad j = 1, 2, \dots, n$$

$$\sum_{i \in S, j \in S} x_{ij} \leq |S| - 1; \quad \forall S \subset V, |S| > 1$$

$$x_{ij} \in \{0,1\}$$

Donde;

n: Número de ciudades a visitar

c_{ij}: Distancia (costo) del nodo *i* al nodo *j*

$$x_{ij} = \begin{cases} 1 & \text{si el arco } (i, j) \text{ está en el TOUR} \\ 0 & \text{si no} \end{cases}$$

2.2.1.1 Comparación entre Métodos de Resolución de Problemas

Para ser más competitivos en el ámbito empresarial, los tiempos en que nos demoramos en ciertas actividades ya sean estas planificaciones de carga, rutas, entre otras; generan pérdidas o re-trabajos por una mala toma de decisiones.

A continuación se muestran dos ejemplos, el primero de ellos se lo trabajó con un nodo de partida y 3 clientes; y al segundo se consideró un cliente adicional. En el primer ejemplo se puede constatar que si se realiza un ruteo de forma empírica “a simple vista”, el tiempo de ejecutar una decisión es igual al tiempo que se demora en realizar un ruteo mediante una metodología; es decir no existe complicación. Sin embargo, para el segundo ejemplo la resolución de este problema se va complicando conforme se añaden más nodos y es ahí donde el utilizar pasos sucesivos como algoritmos, facilita la resolución de este tipo de problemas sin generar duda a quien lo realice.

2.2.1.1.1 Ejemplo No. 1

Para este ejemplo contamos con un vendedor que debe visitar 4 clientes, en donde A es el punto de partida y deberá seguir su recorrido con base a su experiencia; para esto el vendedor ha tomado la decisión de realizar su recorrido o ruteo de forma empírica.

Para la resolución del ejercicio se proporciona las distancias entre puntos a visitar y el grafo inicial, tal como se muestra en la tabla 1 e ilustración 4 respectivamente.

D (i,j)	A	B	C	D
A	-	7	9	8
B		-	10	4
C			-	15
D				-

Tabla 1: Distancias entre Clientes de Ejemplo No. 1

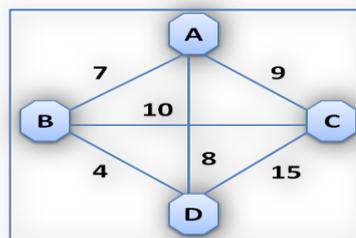


Ilustración 4: Grafo Inicial de Ejemplo No. 1

El vendedor, con su experiencia realiza un ruteo y se pregunta:

¿Cuál es el recorrido en el que me demoro menos en visitar a mis clientes?

Para realizar el análisis empíricamente, en el mejor de los casos realizaría los siguientes pasos:

1. Realizar un listado de caminos posibles que debe tomar.
2. Eliminar los caminos que sean inversos en este caso los que se repiten y generan el mismo costo.
3. Definir que recorrido es para él menor.
4. Regresar al nodo o punto de partida.
5. Finalmente, visitar a los clientes en el orden encontrado.

En el paso 1, el vendedor obtiene los posibles caminos los cuales se detallan a continuación:

A – B – C – D – A	A – B – D – C – A
A – C – B – D – A	A – C – D – B – A
A – D – B – C – A	A – D – C – B – A

En el paso 2, se eliminan los caminos inversos debido a que es lo mismo ir de $A-B$ que de $B-A$; por lo cual el vendedor no considera los siguientes caminos:

A – D – B – C – A
A – C – D – B – A
A – D – C – B – A

En el paso 3, realiza el análisis de recorrido de menor costo, trabajando con los valores de las distancias detalladas en la tabla 1.

$$A - B - C - D - A = 7 + 10 + 15 + 8 = 40$$

$$A - B - D - C - A = 7 + 4 + 15 + 9 = 35$$

$$A - C - B - D - A = 9 + 10 + 4 + 8 = 31$$

En el paso 4, el vendedor por tener pocos clientes se da cuenta que recorre menos yendo con el recorrido: $A - B - D - C - A$ como lo vemos reflejado en el siguiente grafo:

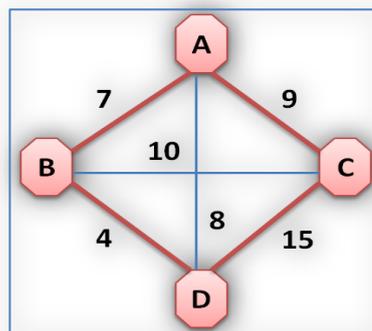


Ilustración 5: Grafo Solución del Ejemplo No. 1

2.2.1.1.1 Aplicación de Algoritmo del Vecino más Cercano para Ejemplo 1

En este apartado se realizó el mismo ejemplo pero con la aplicación de un algoritmo, esta metodología facilitará la resolución del problema en comparación al ejercicio anterior.

Los pasos a seguir para implementar este algoritmo son los siguientes:

1. Identificar cual es el punto de inicio del ejercicio.
2. Desde el nodo de partida se elige cual es el próximo destino comparando las distancias y escogiendo la de menor costo.
3. Repetir el paso 2 hasta recorrer todos los puntos a visitar.
4. Regresar al punto de partida.

Nuestro punto o nodo de partida es "A" y se lo ha coloreado de rojo, tal como se muestra en la ilustración 6.

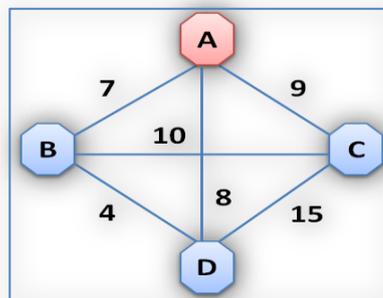


Ilustración 6: Grafo Inicial de Vecino más Cercano Ejemplo No. 1

Como el nodo de partida ya está identificado, se observa que existen tres destinos (aristas) con distancias diferentes, siendo éstas: B, C y D; el algoritmo realiza el análisis y se percata que el destino a seguir es: A – B por tener un recorrido de 7 siendo este menor a las otras dos aristas. Este paso, se lo repite hasta observar que se haya visitado todos los nodos del grafo de este ejercicio, tal como se detalla a continuación en la ilustración 7.

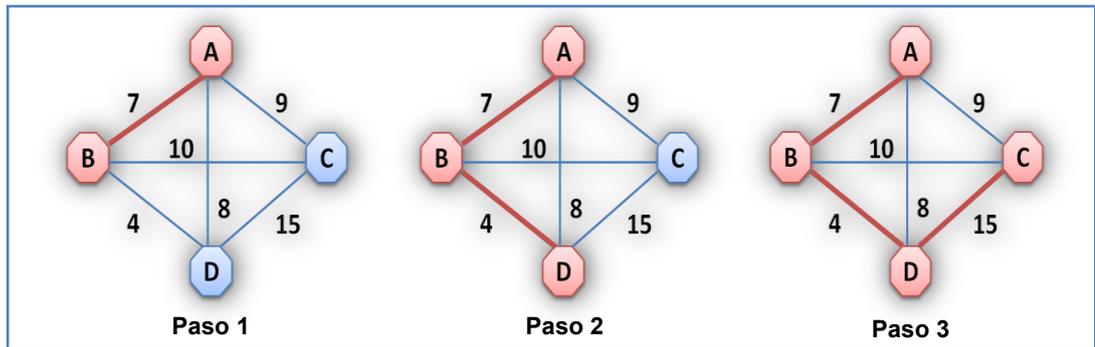


Ilustración 7: Pasos del Algoritmo del Vecino más Cercano del Ejemplo No. 1

Observando la solución final, aplicando el algoritmo nos damos cuenta que es la misma solución del ejercicio realizado de forma empírica; la única diferencia es que mediante esta metodología el tiempo de ejecución del ejercicio puede ser menor.

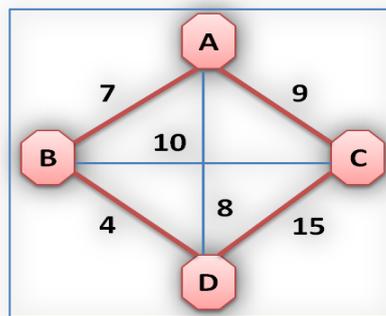


Ilustración 8: Grafo Solución Algoritmo del Vecino más Cercano Ejemplo No. 1

2.2.1.2 Ejemplo No. 2

En este ejemplo un repartidor debe entregar ciertos productos en varias ciudades, las cuales para efecto de estudio se las ha nombrado como: A, B, C, D, E; siendo A el nodo donde el repartidor debe iniciar sus actividades.

De la misma manera que el ejemplo anterior, a continuación se detalla las distancias entre las ciudades y su representación gráfica:

D (i,j)	A	B	C	D	E
A	-	7	9	8	20
B		-	10	4	11
C			-	15	5
D				-	17
E					-

Tabla 2: Distancias entre clientes de Ejemplo No. 2

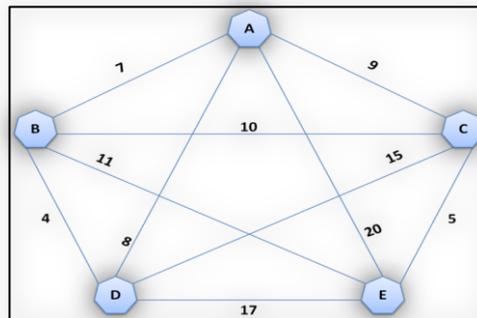


Ilustración 9: Grafo Inicial de Ejemplo No. 2

Realizando el ejercicio de manera empírica, se tuvo que trazar los siguientes caminos posibles:

- | | | |
|-------------|-------------|-------------|
| A-B-C-D-E-A | A-C-D-E-B-A | A-D-E-B-C-A |
| A-B-C-E-D-A | A-C-D-B-E-A | A-D-C-E-B-A |
| A-B-D-C-E-A | A-C-E-D-B-A | A-E-D-B-C-A |
| A-B-D-E-C-A | A-C-E-B-D-A | A-E-D-C-B-A |
| A-B-E-D-C-A | A-D-B-C-E-A | A-E-B-C-D-A |
| A-B-E-C-D-A | A-D-B-E-C-A | A-E-C-B-D-A |
| A-C-B-D-E-A | A-D-C-B-E-A | A-E-B-D-C-A |
| A-C-B-E-D-A | A-D-E-C-B-A | A-E-C-D-B-A |

Luego se eliminaron los caminos inversos, debiendo revisar cada uno de los caminos posibles que no deberían ser tomados en cuenta, obteniendo:

- | | | |
|-------------|-------------|--------------|
| A-B-C-E-D-A | A-B-E-C-D-A | A-C-B-E-D-A |
| A-C-D-E-B-A | A-C-E-D-B-A | A-C-E-B-D-A |
| A-E-D-B-C-A | A-E-D-C-B-A | A-E-B-C-D-A |
| A-E-C-B-D-A | A-E-B-D-C-A | A-E-C-D-B-A; |

Se realizó el análisis de recorrido de menor costo, luego de haber trabajado con la tabla 2 de distancia entre los clientes a ser visitados por nuestro repartidor.

$A - B - D - C - E - A = 7+4+15+5+20$	$=$	51
$A - B - D - E - C - A = 7+4+17+5+9$	$=$	42
$A - B - C - D - E - A = 7+10+15+17+20$	$=$	69
$A - B - E - D - C - A = 7+11+17+15+9$	$=$	59
$A - D - C - B - E - A = 8+15+11+10+20$	$=$	64
$A - D - C - E - B - A = 8+15+5+11+17$	$=$	46
$A - D - E - B - C - A = 8+17+11+10+9$	$=$	55
$A - D - E - C - B - A = 8+17+5+10+7$	$=$	47
$A - D - B - C - E - A = 8+4+10+5+20$	$=$	47
$A - D - B - E - C - A = 8+4+11+5+9$	$=$	37
$A - C - D - B - E - A = 9+15+4+11+20$	$=$	59
$A - C - B - D - E - A = 9+10+4+17+20$	$=$	60

En este caso nuestro repartidor de productos escoge el recorrido **A-E-C-D-B-A** dando como resultado del recorrido total de 37 Km. En el siguiente grafo se puede apreciar el resultado obtenido.

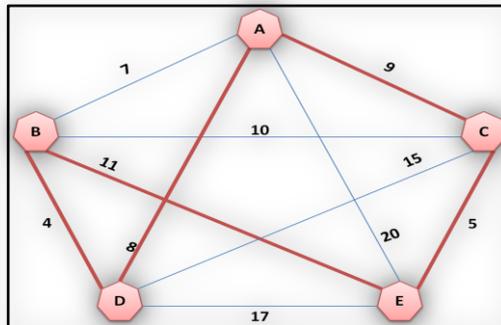


Ilustración 10: Grafo Solución del Ejemplo No. 2

2.2.1.2.1 Aplicación del Algoritmo del Vecino más Cercano para el Ejemplo 2

De la misma manera que en el ejemplo 1, se aplicó el algoritmo del Vecino más Cercano, definiendo como el punto de partida a "A", recordando que el recorrido termina cuando volvemos al mismo punto.

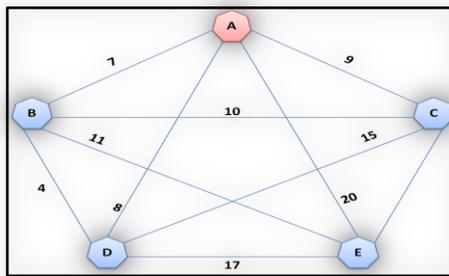


Ilustración 11: Grafo Inicial de Vecino más Cercano Ejemplo No. 1

Luego de esto, se procede a observar cual de los siguientes clientes va a ser visitado. Desde A podemos visitar: A-B; A-C; A-D; A-E, de estas alternativas el algoritmo escoge la distancia de menor costo, seleccionando el recorrido A-B, repitiendo esta operación hasta recorrer todos los nodos y llegar al punto inicial.

A continuación en la ilustración 12 se presenta de manera gráfica los pasos que realiza este algoritmo hasta llegar a la solución final.

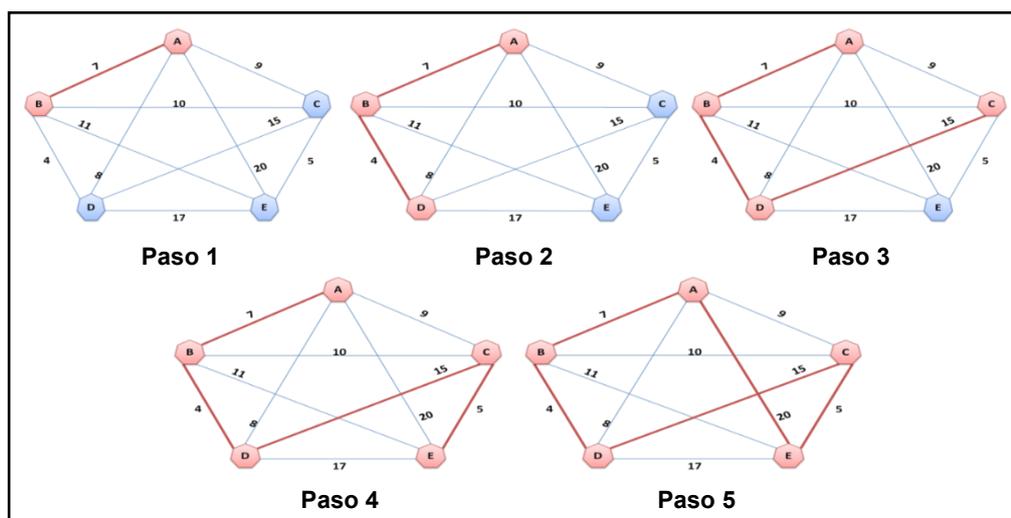


Ilustración 12: Pasos del Algoritmo del Vecino más Cercano del Ejemplo No. 2

Es importante mencionar que este método a pesar de que genera un camino más corto no siempre es el ideal, es por ello que el algoritmo del Vecino más cercano debe ser asociado con otras heurísticas o realizar cambios dentro de la programación para obtener los resultados deseados.

Por consiguiente, para realizar el ruteo vehicular de la empresa de nuestro estudio, se decidió adaptar el algoritmo de Vecino más cercano al problema de ruteo vehicular con ventanas de tiempo VRPTW e implementar un algoritmo de mejora como el Recocido Simulado, para de esta manera encontrar de un conjunto de soluciones factibles la mejor.

2.2.2 Definición Matemática del VRPTW

El problema del Ruteo Vehicular con Ventanas de Tiempo (VRPTW), está definido matemáticamente de la siguiente manera [1]:

Datos de entrada:

Un grafo dirigido $D = (U \cup V, A)$ con:

- $U = \{u_1, u_2, \dots, u_m\}$ (depósitos)
- $V = \{v_1, v_2, \dots, v_n\}$ (clientes)

Los nodos en U representan los depósitos con capacidades $k_1, k_2 \dots, k_m$ (número de vehículos).

Para elaborar nuestro proyectos, se trabajó con un depósito (u_1), que es la sucursal ubicada en la ciudad de Milagro.

- Cada depósito $u \in U$ contiene una flota homogénea de vehículos, cuyos costos (tiempos) de desplazamiento están dados por un vector $c^u \in \mathfrak{R}^{V \times V}$.
- Los nodos en V representan clientes y tienen asociados tiempos de servicio $\delta_v, \forall v \in V$.
- Asociado a cada cliente $v \in V$ está además una ventana de tiempo $[a_v, b_v]$ en la cual la visita a v debe empezar.

Formalización:

Costos de desplazamiento modificados:

$$\tilde{C}_{w\bar{w}}^u = \begin{cases} C_{w\bar{w}}^u + \delta_{\bar{w}}; & \text{si } \bar{w} \in V \\ C_{w\bar{w}}^u; & \text{caso contrario} \end{cases}$$

Compatibilidad entre nodos cliente: (v, \bar{v}) , es un par ordenado de clientes compatibles, si algún vehículo puede visitar \bar{v} inmediatamente después de v , es decir, si:

$$\exists u \in U: a_v + \tilde{C}_{w\bar{w}}^u \leq b_{\bar{v}}$$

Arcos:

Tres tipos de arcos:

$$A_s = \{(u, v) : u \in U, v \in V\} \quad (\text{salida})$$

$$A_r = \{(u, v) : u \in U, v \in V\} \quad (\text{retorno})$$

$$A_c = \{(v, \bar{v}) : v, \bar{v} \in V, (v, \bar{v}) \text{ es compatible}\} \quad (\text{conexión})$$

Rutas:

Una ruta r es un circuito cerrado dirigido que contiene exactamente un nodo de $u \in U$:

$$r = (u, \bar{v}_1, \bar{v}_2, \dots, \bar{v}_k, u)$$

Con $\bar{v}_1, \bar{v}_2, \dots, \bar{v}_k \in V$

Rutas factibles:

Una ruta es factible si cada nodo cliente es visitado dentro de su ventana de tiempo:

$$a_{\bar{v}_j} \leq \tilde{C}_{u\bar{v}_1}^u + \sum_{s=1}^{j-1} \tilde{C}_{\bar{v}_s\bar{v}_{s+1}}^u \leq b_{\bar{v}_j}, \forall j = 1, \dots, k.$$

Costo de rutas:

El costo $c(r)$ (o duración) de r esta dado por:

$$c(r) = \tilde{C}_{u\bar{v}_1}^u + \sum_{s=1}^{k-1} \tilde{C}_{\bar{v}_s\bar{v}_{s+1}}^u + \tilde{C}_{\bar{v}_k u}^u$$

Problema:

Encontrar un conjunto de rutas factibles tal que:

- Cada cliente sea atendido por una sola ruta dentro de su ventana de tiempo.
- El costo acumulado de las rutas sea mínimo.

Variables de decisión:

$$X_{w\bar{w}}^u = \begin{cases} 1; & \text{si } (w, \bar{w}) \text{ es usado en alguna ruta de } u \in U \\ 0; & \text{caso contrario} \end{cases}$$

Tiempos de visita:

$\forall v \in V: T_v \in \mathfrak{R}$: Tiempo en que el cliente v es visitado.

Todas estas variables previamente declaradas, formaron parte del modelo matemático final, el mismo que luego será llevado a un lenguaje de programación, respetando todas sus funciones y estructura, la que asegura la correcta función de nuestro software.

Por medio de esto, se da a conocer, que el modelo matemático es parte fundamental y vital de todo nuestro programa.

2.3.1 Modelo Matemático del VRPTW**Función Objetivo:**

$$\min \sum_{u \in U} \sum_{(w, \bar{w}) \in A} \tilde{C}_{w\bar{w}}^u \tilde{X}_{w\bar{w}}^u \quad (1)$$

Restricciones:

$$\sum_{u \in U} \sum_{(v, w) \in A} X_{vw}^u = 1; \quad \forall v \in V \quad (2)$$

$$\sum_{(v, w) \in A} X_{vw}^u = \sum_{(w, v) \in A} X_{wv}^u; \quad \forall v \in V, \forall u \in U \quad (3)$$

$$\sum_{v, \bar{v} \in W} X_{v\bar{v}}^u \leq |W| - 1; \quad \forall W \subseteq V, \forall u \in U \quad (4)$$

$$\sum_{(u,v) \in A} X_{uv}^{\bar{u}} + \sum_{(v,u) \in A} X_{vu}^{\bar{u}} \leq 0; \quad \forall u, \bar{u} \in U, u \neq \bar{u} \quad (5)$$

$$\sum_{(u,v) \in A} X_{uv}^u \leq \kappa_u; \quad \forall u \in U \quad (6)$$

$$a_v \leq T_v \leq b_v; \quad \forall v \in V \quad (7)$$

$$X_{v\bar{v}}^u (T_v + \tilde{C}_{v\bar{v}}^u - T_{\bar{v}}) \leq 0; \quad \forall (v, \bar{v}) \in A_c, \forall u \in U \quad (8)$$

$$X_{uv}^u (\tilde{C}_{uv}^u - T_v) \leq 0; \quad \forall (u, v) \in A_s, \forall u \in U \quad (9)$$

$$X_{w\bar{w}}^u \in \{0,1\}; \quad \forall (w, \bar{w}) \in V, \forall u \in U$$

$$T_v \in \mathfrak{R}; \quad \forall v \in V$$

Donde:

- (1) Función objetivo, Minimizar la distancia total recorrida, corresponde a la suma de los costos de los arcos utilizados.
- (2) Asegura que todo cliente que haya realizado el pedido deberá ser visitado.
- (3) Restricción de equilibrio de flujo de vehículos, indica que la ruta debe llegar y abandonar una sola vez cada cliente.
- (4) Restricción de eliminación de sub-tours, elimina un ciclo cuando se tiene el número de arcos menor al número de vértices que existe.
- (5) Esta restricción evita que el depósito sea considerado como un cliente y no sea visitado inmediatamente.
- (6) Restricción de capacidad, indica la ruta establecida no deberá ser mayor a la capacidad del número de vehículos del depósito.
- (7) Restricción de ventana horaria, asegura que se cumpla el tiempo de servicio de cada cliente entre el intervalo de tiempo establecido.
- (8) Y (9) Estas restricciones, influyen para que se cumpla los tiempos de recepción de los clientes dentro de las ventanas horarias establecidas, respetando el orden de las mismas.

2.3.2 Complejidad Computacional del VRPTW

Los problemas de optimización de rutas y scheduling son de gran complejidad si miramos el recurso tiempo de ejecución. Podemos definir un algoritmo como un procedimiento paso a paso para resolver un problema. Uno de los objetivos cuando se diseña un algoritmo es el de encontrar el método más eficiente, midiéndose dicha eficiencia en tiempo de ejecución. Este tiempo depende del tamaño de la entrada del problema, por tanto el coste de un problema, en términos de tiempo de ejecución, se mide mediante una función que relaciona el tamaño de la entrada con el tiempo. Así los problemas de optimización de rutas de vehículos se pueden clasificar dentro de los denominados NP-duros, los cuales podríamos definirlos a grandes rasgos diciendo que el esfuerzo computacional requerido para resolverlos aumenta exponencialmente con el tamaño de la entrada. Como el VRPTW engloba diferentes problemas de optimización conocidos que son NP-duros, podemos decir que el VRPTW es NP-duro en sentido fuerte [4].

2.4 Algoritmo de mejoramiento Recocido Simulado (Simulated Annealing).

Simulated Annealing (SA) o recocido simulado es un algoritmo de búsqueda meta-heurística para problemas de optimización global; el objetivo general de este tipo de algoritmos es encontrar una buena aproximación al valor óptimo de una función en un espacio de búsqueda grande. A este valor óptimo se lo denomina "óptimo global".

El nombre e inspiración viene del proceso de recocido del acero y cerámicas, una técnica que consiste en calentar y luego enfriar lentamente el material para variar sus propiedades físicas. El calor causa que los átomos aumenten su energía y que puedan así desplazarse de sus posiciones iniciales (un mínimo local de energía); el enfriamiento lento les da mayores probabilidades de recristalizar en configuraciones con menor energía que la inicial (mínimo global).

El método fue descrito independientemente por Scott Kirkpatrick, C. Daniel Gelatt y Mario P. Vecchi en 1983 y por Vlado Cerný en 1985. El método es una adaptación del algoritmo Metrópolis-Hastings, un método de Montecarlo utilizado para generar muestras de estados de un sistema termodinámico [9].

2.4.1 Iteración básica

En cada iteración, el método de recocido simulado evalúa algunos vecinos del estado actual s y probabilísticamente decide entre efectuar una transición a un nuevo estado s' o quedarse en el estado s . En el ejemplo de recocido de metales descrito arriba, el estado s se podría definir en función de la posición de todos los átomos del material en el momento actual; el desplazamiento de un átomo se consideraría como un estado vecino del primero en este ejemplo. Típicamente la comparación entre estados vecinos se repite hasta que se encuentre un estado óptimo que minimice la energía del sistema o hasta que se cumpla cierto tiempo computacional u otras condiciones [9].

2.4.2 Vecindario de un estado

El vecindario de un estado s está compuesto por todos los estados a los que se pueda llegar a partir de s mediante un cambio en la conformación del sistema. Los estados vecinos son generados mediante métodos de Montecarlo.

El método de evaluación de estados vecinos es fundamental para encontrar una solución óptima global al problema dado. Los algoritmos heurísticos, basados en buscar siempre un estado vecino mejor (con energía más baja) que el actual se detienen en el momento que encuentran un mínimo local de energía. El problema con este método es que no puede asegurar que la solución encontrada sea un óptimo global, pues el espacio de búsqueda explorado no abarca todas las posibles variaciones del sistema [9].

2.4.3 Probabilidad de Transición

La probabilidad de hacer la transición al nuevo estado s es una función $P(\delta E, T)$ de la diferencia de energía $\delta E = E(s') - E(s)$ entre los dos estados, y de la variable T , llamada temperatura por analogía con el concepto físico de temperatura.

Si δE es negativo, es decir, la transición disminuye la energía, el movimiento es aceptado con probabilidad $P = 1$. Es importante remarcar que la condición de que el sistema siempre pase a un sistema de menor energía cuando se encuentra una no es en absoluto necesaria para el éxito del método. Cuando δE es positivo la

probabilidad de transición P es siempre distinta de cero; es decir, el sistema puede pasar a un estado de mayor energía (peor solución) que el estado actual. Esta propiedad impide que el sistema se quede atrapado en un óptimo local.

A medida que la temperatura tiende al mínimo, la probabilidad de transición a un estado de mayor energía tiende a cero asintóticamente. Cuando T llega a cero, el algoritmo solo aceptará cambios a estados con menor energía. Debido a esta propiedad, la temperatura juega un papel muy importante en el control de la evolución del sistema. A temperaturas altas, el sistema tenderá a saltos de energía grandes entre los estados, mientras que a temperaturas más bajas, los cambios en energía serán menores.

Así, en cada iteración el algoritmo tiende a encontrar estados con menor energía total. Hay muchas maneras de disminuir la temperatura, siendo la más usual la exponencial, donde T disminuye por un factor $\alpha < 1$ en cada paso [9].

2.4.4 Protocolo de recocido

Como el nombre del algoritmo sugiere, la variación de la temperatura durante la computación es una característica distintiva de este método. El algoritmo comienza con un valor de T muy alto, que va decreciendo en cada iteración siguiendo un cierto protocolo de recocido, que puede ser diferente para cada problema, pero que siempre debe terminar con $T = 0$. Así el sistema será libre inicialmente de explorar una gran porción del espacio de búsqueda, ignorando pequeñas variaciones de la energía entre los estados vecinos evaluados, para más tarde centrarse en regiones con estados de baja energía y, al final, cambiar solo a estados con energía menor que la inicial, hasta alcanzar un mínimo.

La probabilidad de que el algoritmo acabe encontrando el mínimo global para un problema dado se aproxima a 1 a medida que el protocolo de recocido se extiende [9].

3. Elaboración del Programa

3.1 Selección del Software de Programación

Hoy en día, existen diferentes software de programación que son herramientas que permiten simular mediante varios lenguajes tales como C, C++, C#, JAVA, PHP, etc. las distintas necesidades del programador de una manera más práctica y eficaz.

De acuerdo a los distintos programas utilizados durante las etapas de nuestra vida académica y a la experiencia con los diferentes software que ayudaron en la optimización de recursos con bajos tiempos de ejecución. Se consideró la utilización del software **Mathematica 7.0** para la elaboración de este proyecto, que permitió la codificación del algoritmo del vecino más cercano con un algoritmo de mejora recocido simulado, siendo éste una aplicación muy útil y compatible con varios programas para la extracción de datos y a su vez la exportación de información y/o resultados; así como también entre muchas de sus funciones, la facilidad de la representación gráfica, contando con una gran variedad de funciones preestablecidas que permiten de manera más sencilla crear y desarrollar las distintas necesidades de los usuarios.

3.2 Estructura de los Datos

Para la elaboración de nuestro programa se utilizó el software, Mathematica 7.0, posee una capacidad la cual consiste en la importación de archivos externos siendo esta una ventaja para el usuario, por no tener que ingresar manualmente los datos en el programa almacenándolos e importándolos en una base de datos, por ello con esta ventaja se consideró definir una estructura adecuada de los datos a ser importados y así asegurar el correcto funcionamiento del mismo.

Por consiguiente, hemos decidió generar una base de datos en el programa Microsoft Office Access 2007, el cual sirve para la creación y control de informes datos, de forma rápida y sencilla, gracias a la interfaz de usuario de Microsoft Office Fluent y funciones de diseño interactivas que no requieren conocimientos especializados de bases de datos. Y lo convierten en una herramienta amigable para los usuarios.

La base de datos creada en Access, contiene el nombre del cliente, las coordenadas que fueron obtenidas utilizando el sistema informático Google Earth, la demanda asociada y tiempos de recepción de productos para de esta manera importar los datos de Access al software Mathematica mediante un comando de conexión y generar la solución.

A continuación en la tabla 3 se muestra el formato del ingreso de la información en la base de datos:

c_i	$coordx_i$	$coordy_i$	ti_i	tf_i	ts_i
c_1	$coordx_1$	$coordy_1$	ti_1	tf_1	ts_1
c_2	$coordx_2$	$coordy_2$	ti_2	tf_2	ts_2
.
.
.
c_{n-1}	$coordx_{n-1}$	$coordy_{n-1}$	ti_{n-1}	tf_{n-1}	ts_{n-1}
c_n	$coordx_n$	$coordy_n$	ti_n	tf_n	ts_n

Tabla 3: Estructura de ingreso de datos

Donde:

- c_i = Cliente; $i = 1, \dots, n$
- $coordx_i$ = Coordenada x del cliente i ; $i = 1, \dots, n$
- $coordy_i$ = Coordenada y del cliente i ; $i = 1, \dots, n$
- ti_i = Tiempo de inicio de la ventana de tiempo del cliente i ; $i = 1, \dots, n$
- tf_i = Tiempo de fin de la ventana de tiempo del cliente i ; $i = 1, \dots, n$
- ts_i = Tiempo de servicio otorgado al cliente i ; $i = 1, \dots, n$

Con esa edición en la base de datos, el programa generara la ruta óptima, para el despacho a los clientes que pusieron requerimientos de mercadería, por los diferentes canales de recepción de la empresa.

3.3 Funciones

Para facilitar la programación del problema abordado, se crearon diferentes funciones que ayudan en la codificación y ejecución; las cuales se detallan a continuación:

3.3.1 Función Crear Coordenada de Cliente

Esta función recibe las coordenadas x y y de cada uno de los clientes de la empresa cuya información fue obtenida con la ayuda del sistema informático Google Earth.

ClientesCoord[coordx, coordy]

3.3.2 Función Grafo Cliente.

Esta función nos permite observar la representación gráfica de cada uno de los clientes de la empresa mediante un grafo.

GrafoClientes[aristas, clientescoord]

3.3.3 Función Crear Matriz de Distancias

Mediante esta función se calcula la distancia de un cliente en particular con respecto al resto, utilizando la formula de distancia $d_{ij} = |x_j - x_i| + |y_j - y_i|$ correspondiente a la *Métrica Manhattan*.

CrearMatrizDistancias[datos]

3.3.4 Función Vecino Más Cercano

Mediante esta función generaremos a cada cliente un ‘vecino’ el que se encuentre más cercano para realizar la próxima visita dentro de la programación y que sea atendido; este paso se repetirá hasta culminar la visita a todos los clientes de la empresa, es muy importante asignar una bandera dentro de esta función para que no exista repetición en la visita de los clientes.

VecinoMasCercano[matriz, nodo, band]

3.3.5 Función Crear Ruta

Esta función creará y guardará la ruta de los clientes ya visitados y aquel cliente que se encuentre más lejano no se lo tomará en cuenta hasta que un vecino esté cerca de él. Así mismo, dentro de esta función se implementará una bandera para aislar al cliente que haya sido visitado, para no tomarlo en cuenta dos veces.

CrearRuta[datos, matriz, band]

3.3.6 Función Crear Rutas Iniciales

Esta función se encarga de crear las rutas que se generan mediante la recepción de pedidos y así atender a los clientes que hayan realizado el pedido en el día.

CrearRutasIniciales[datos, matriz]

3.3.7 Función Graficar Solución

Mediante esta función se genera el gráfico de las rutas creadas, donde se puede observar cual cliente es visitado primero y los turnos del resto de clientes que serán establecidos por el programa.

GraficarSolucion[datos, rutas]

3.3.8 Función Calcular Costo Ruta

Esta función calcula el costo de cada una de las rutas posibles, para luego unir con el resto de rutas creadas mediante los pedidos de los clientes.

CalcularCostoRuta[matriz, ruta]

3.3.9 Función Calcular Costo Ruteo

Con la ayuda de esta función tendremos el costo total del recorrido; es decir, la suma del costo de las rutas creadas, escogiendo la de menor costo optimizando las distancias recorridas que impactan directamente en la disminución de los costos.

CalcularCostoRuteo[matriz, rutas]

3.3.10 Función Scheduling Ruta

Esta función permite crear el horario en el que el cliente puede ser visitado por el camión de la empresa; respetando la ventana de tiempo.

SchedulingRuta[datos, matriz, ruta]

3.3.11 Función Calcular Penalización

Esta función sirve para penalizar al transportista por irrespetar la ventana de tiempo; es decir por llegar fuera del horario establecido de atención.

CalcularPenalizacion[datos, scheduling]

3.3.12 Función Costo Total Ruta

Una vez calculado el costo de cada una de las rutas en el punto 4.2.8 de este apartado en esta función se suman los costos de penalización asociados.

CostoTotalRuta[datos, matriz, ruta]

3.3.13 Función Costo Total Ruteo

De la misma manera que la función anterior se incorporará los costos de penalización al recorrido total.

CostoTotalRuteo[datos, matriz, rutas]

3.3.14 Función Generar Vecino

La finalidad de esta función, es de escoger del conjunto de rutas generadas la de menor costo posible, respetando las ventanas de tiempo y visitando todos los clientes tomando en consideración todos los costos asociados al recorrido total, siendo esta una de las funciones más importantes de nuestra programación.

GenerarVecino[rutas]

3.5 Programa Principal

Para el desarrollo de nuestro proyecto se consideró realizar como programación principal un algoritmo de búsqueda local conocido como Recocido Simulado (*Simulated Annealing*); para ello la creación de las funciones descritas en el punto anterior que permitieron obtener la grafica de los clientes, el cálculo de las rutas, costos y la generación del algoritmo del vecino más cercano obteniendo una solución inicial, para a partir de este resultado implementar el Recocido Simulado optimizando distancias, costos y tiempo de servicio.

Aplicando este programa, se logra satisfacer los requerimientos de los consumidores, evitando tengan que recurrir a otras empresas para obtener sus productos, que es lo que no se debe permitir bajo ninguna circunstancia.

Nuestro programa principal se basa en la naturaleza del Recocido Simulado y está constituido por los siguientes pasos:

- Importar los datos desde archivos externos.
- Crear una matriz de distancias.
- Crear la solución inicial del problema usando el algoritmo Vecino mas cercano y llamarla solución actual.
- Crear una solución vecina a la solución actual intercambiando la posición de dos nodos cualesquiera en una misma ruta cualquiera de la solución actual.
- Recalcular las ventanas de tiempo con las fórmulas anteriores.
- Calcular el valor de la función objetivo.
- Si la nueva solución es mejor que la solución actual, aceptarla inmediatamente; caso contrario, generar un número aleatorio uniformemente distribuido entre 0 y 1, si este número aleatorio es menor o igual a la probabilidad de Boltzman aceptar la nueva solución, caso contrario rechazarla y continuar con la siguiente iteración.

Entendiendo como probabilidad de Boltzman a $e^{-\Delta/T}$; donde:

$$\Delta = \text{costo}(x) - \text{costo}(y)$$

$T = \text{Temperatura}$; Parámetro de calibración.

- Graficar la ruta más optima encontrada.

A continuación se presenta el pseudocódigo de lo antes descrito:

Inicio

Importar (datos)

Distancia = CrearMatrizDistancias (Datos)

SolIni = CrearRutasInicilaes (Datos, Dist)

CostIni = CostoTotalRuteo (Datos, Dist, SolIni)

Para i = 1 hasta todos los clientes

SolNew = GenerarVecino(SolIni)

CostNew = CostoTotalRuteo(Datos, Dist, SolNew)

Si CostNew es menor que CostIni

SolIni = SolNew; CostIni = CostNew

caso contrario

NumAleatorio = NumAleatorio {0,1}

ProBolzman = Calcula Probabilidad de Bolzman

Si NumAleatorio es menor igual a ProBolzman

SolIni = SolNew; CostIni = CostNew

Fin Si

Fin Si

Fin Para

Mostrar Recorrido de Costo Minimo

Graficar Recorrido de Costo Minimo (SolIni)

Fin

4. Desarrollo y Análisis del Problema

Para el desarrollo del problema, hemos considerado que la frecuencia de recepción de pedidos de los clientes sea diariamente; es decir, cada día se atiende a un subconjunto del total de clientes, esto se debe a que la demanda de la empresa es aleatoria. Por consiguiente, siempre se reciben pedidos de máximo siete clientes por ruta y esto genera realizar de tres a cuatro rutas diarias, asegurando un excelente servicio al cliente como política de la empresa.

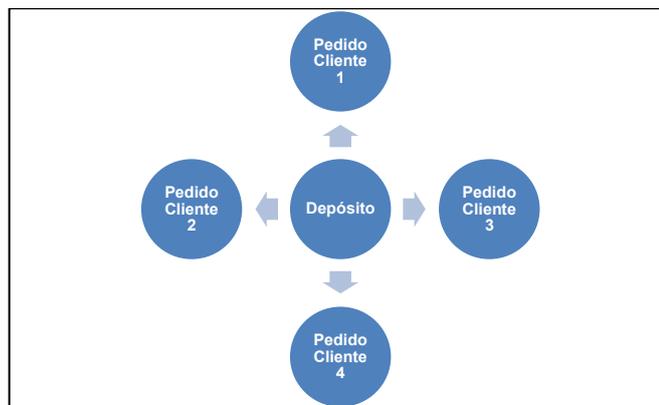


Ilustración 13: Función de Distribución Actual de Empresa

4.1 Tablas de Información

En la tabla 4 que se muestra a continuación, se encuentra el detalle de los parámetros utilizados para el desarrollo de nuestro proyecto; es decir la información que alimenta nuestra programación, contando con los siguientes datos:

- *Cientes*, a los cuales se les ha asignado un número; donde el depósito esta denominado como cliente 1, teniendo un total 36 clientes incluyendo al depósito, que por motivos de confidencialidad no se menciona los nombres.
- *X y Y*, corresponde a las coordenadas X y Y de cada cliente que se obtuvieron para calcular las distancias entre ellos.
- *T_i y T_f*, corresponde a las ventanas de tiempo inicial y final de los clientes.
- *T_{ser}*, Los tiempos de servicio es el mismo para todos los clientes, teniendo en cuenta una holgura para cualquier tipo de imprevisto, ya que los clientes suelen estar ocupados, pero no significa, que no se le pueda realizar la entrega.

Ciente	X	Y	Ti	Tf	Tser
1	0	0	0	300	0
2	-4380	-6018	912	1212	45
3	-803	-2002	825	1125	45
4	-3034	-7082	65	365	45
5	-4915	-10580	727	1027	45
6	18219	-7745	15	315	45
7	-4378	-9207	621	921	45
8	27317	-30667	170	470	45
9	-14377	-18909	255	555	45
10	-14965	-19703	534	834	45
11	2480	-13136	357	657	45
12	15594	-20838	448	748	45
13	15497	-21159	652	952	45
14	38552	-18799	30	330	45
15	21965	-21153	567	867	45
16	24134	-23570	384	684	45
17	27079	-23869	475	775	45
18	32823	-20454	99	399	45
19	30694	-22101	179	479	45
20	46917	-17980	278	578	45
21	27582	-31270	10	310	45
22	27936	-31369	914	1214	45
23	28264	-31210	812	1112	45
24	28721	-31397	732	1032	45
25	28713	-31881	65	365	45
26	12113	3741	169	469	45
27	17129	-2729	357	657	45
28	14759	-3593	448	748	45
29	4106	-1678	652	952	45
30	40098	3236	30	330	45
31	38220	-1123	567	867	45
32	33658	-545	384	684	45
33	44191	-5236	475	775	45
34	31873	-22795	99	399	45
35	32918	-11606	914	1214	45
36	32263	-22813	812	1112	45

Tabla 4: Tabla de Distancias de Clientes

5. Resolución del Problema.

Las distancias actuales recorridas por el camión de la empresa, son confidenciales. Pero se conoce que actualmente una de las políticas de la empresa es entregar al cliente que pide primero, siendo en este caso el primer destino a visitar y así sucesivamente, sin considerar que estos se encuentren muy alejados el uno del otro, generándose subciclos y a su vez alargando el recorrido, esto sucede porque cuentan con un sistema de ruteo que solo lo hace el chofer de dicho vehículo a su propia conveniencia.

Luego de este breve análisis, podemos concluir que las distancias recorridas diariamente por este sistema de distribución, son extensas.

Mediante un comando de programación, se conectó la base de datos de los clientes a nuestro programa, enlazando la información de entrada para que este genere la solución. Esto se lo hace con la finalidad de optimizar tiempo y facilitar la ejecución, ya que si se realiza algún cambio en la base de datos de Access, esta automáticamente se actualiza, generando la nueva solución en el programa.

Ahora se muestra un gráfico que representará el recorrido de la empresa realizado de manera empírica; es decir se simula una posible instancia de recorrido que no considera ninguna restricción asociada a los clientes; como ventanas de tiempo, distancias, etc.; sino las políticas antes mencionados, todo esto con el fin de realizar una comparación con el algoritmo de mejora realizado en este proyecto.

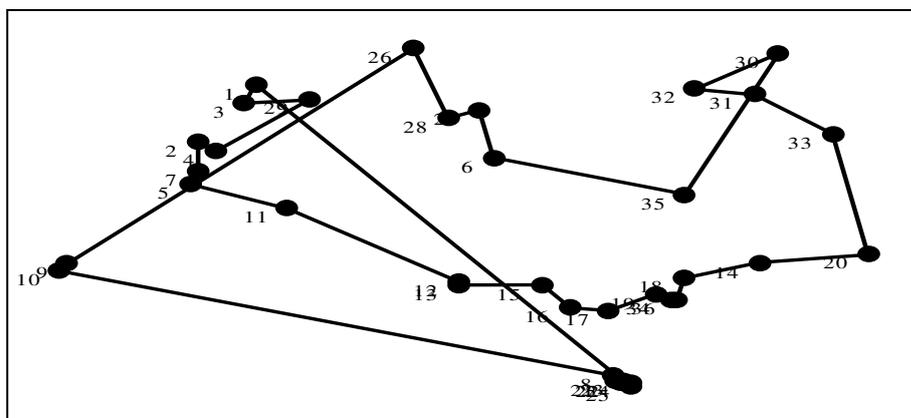


Ilustración 14: Ruteo Empírico

La distancia total recorrida de esta ruta es: 555.2659 *km*

Siendo el recorrido los nodos de la siguiente manera:

{1, 3, 29, 4, 2, 7, 5, 11, 12, 13, 15, 16, 17, 19, 34, 36, 18, 14, 20, 33, 31, 32, 30, 35, 6, 27
, 28, 26, 9, 10, 8, 21, 22, 23, 24, 25, 1}}

Por otro lado, una vez obtenido el resultado del recorrido empírico, se procedió a aplicar el algoritmo de mejora programado para este problema en particular y así comparar resultados.

A continuación se detalla cada una de las rutas generadas de nuestro programa para de esta manera ver el comportamiento y la disminución de la distancia recorrida y así evaluar la calidad de los resultados que éste genera:

Este algoritmo genera 4 rutas factibles, las detallamos a continuación:

Ruta 1:

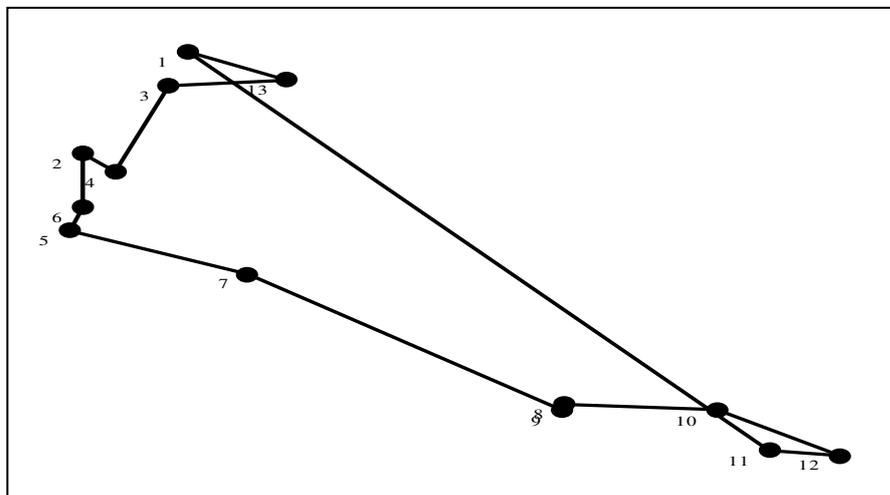


Ilustración 15: Ruta 1

Ruta 2:

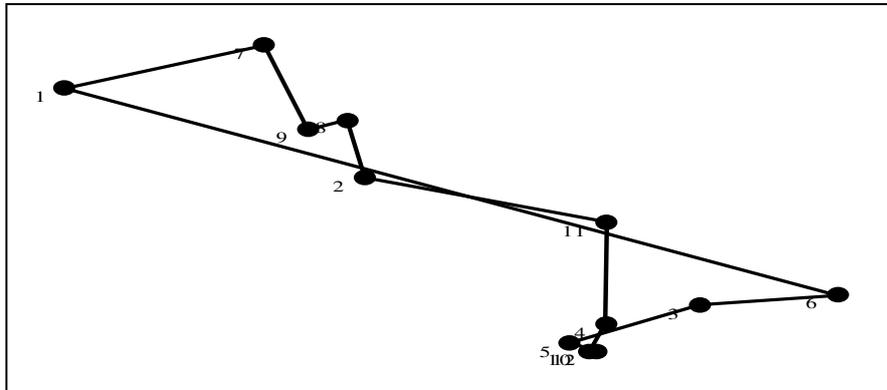


Ilustración 16: Ruta 2

Ruta 3:

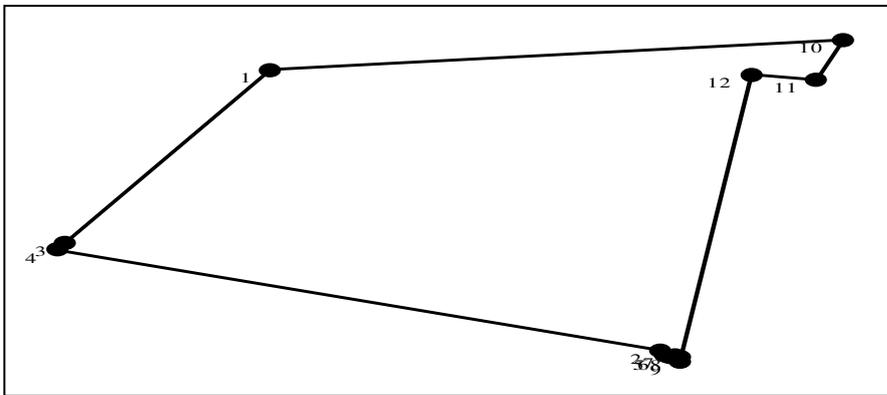


Ilustración 17: Ruta 3

Ruta 4:

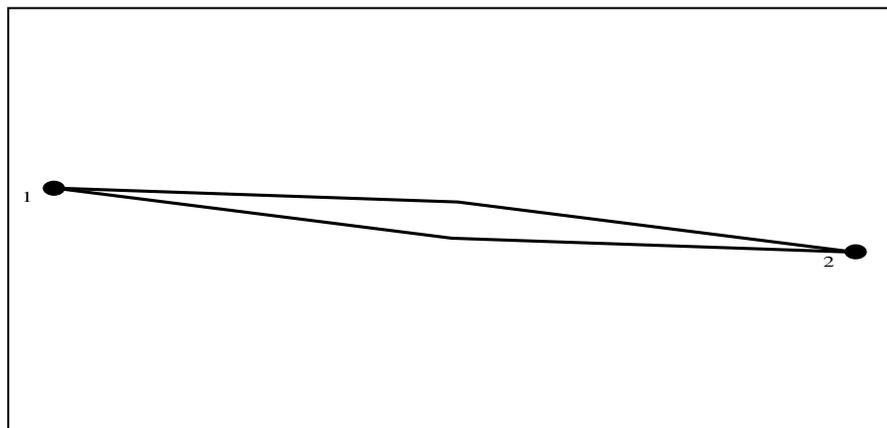


Ilustración 18: Ruta 4

Cada una de estas rutas representara el recorrido que genera el programa, regresando cada vez al centro de distribución llamado depósito.

Una vez generadas cada una de las rutas, el programa procede a unir las mismas, detallando el recorrido total que el chofer de la empresa deberá respetar siguiendo los parámetros establecidos en la solución final del programa que se muestra a continuación en la ilustración 19.

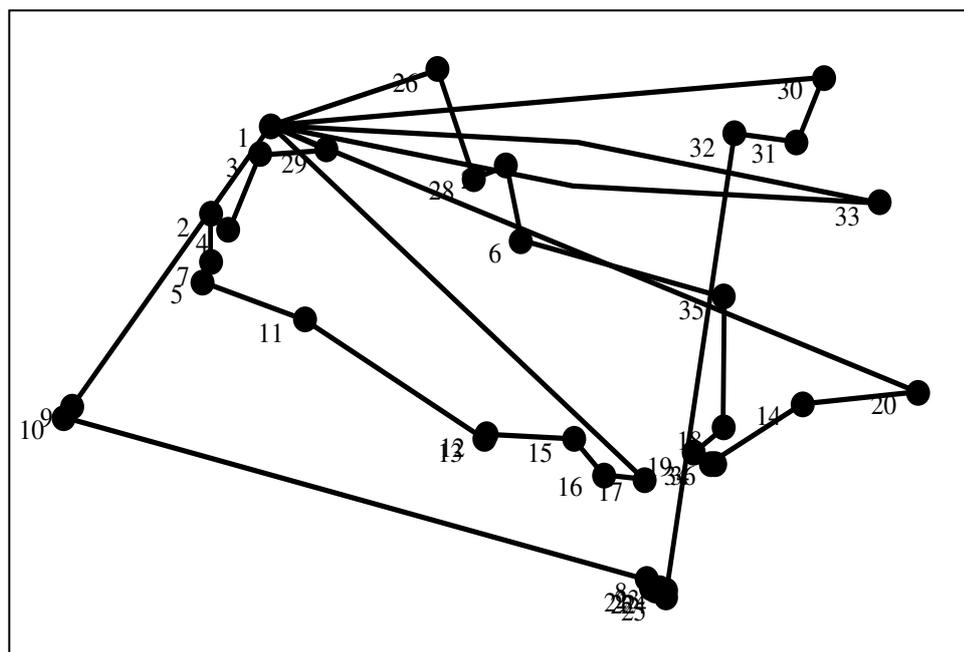


Ilustración 19: Ruteo con el Algoritmo de recocido Simulado

La distancia total recorrida en las nuevas rutas es: 320.7908 km

Las nuevas rutas están comprendidas por el siguiente conjunto:

$$\{\{1,29,3,4,2,7,5,11,13,12,15,16,17,1\}, \{1,26,28,27,6,35,18,19,36,34,14,20,1\},$$

$$\{1,9,10,8,21,22,23,25,24,32,31,30,1\}, \{1,33,1\}\}$$

Como se puede observar, existe un significativo ahorro de distancia recorrida en comparación con la instancia empírica generada, logrando una reducción del 42% de la distancia total recorrida correspondiente a 234.4751km menos, impactando directamente al tiempo total de recorrido y a su vez a los costos directos e indirectos

asociados a esta operación como lo es el combustible, mantenimiento, entre otros. Pero sobre todo satisfaciendo a los clientes al realizar entregas mejor programadas.

Cada una de estas rutas, está asociada con un Scheduling, que es el que establece los tiempos de llegada, tiempos de salida y tiempos de servicio, que son los tiempos que el camión deberá asumir para atender a cada uno de los clientes establecidos en las rutas.

Todos estos tiempos fueron previamente establecidos, por políticas de empresa, y actualizados por principios logísticos, conociendo que para toda empresa el tiempo significa dinero y a medida que los tiempos se disparan se pierden oportunidades y se generan gastos innecesarios para la compañía.

A continuación, se detallan cada una de las tablas con los respectivos Scheduling o rutas de los resultados de la programación realizada:

Scheduling de Ruta 1:

Cliente	Tiempo Llegada	Tiempo de Servicio	Tiempo de Salida
1	0	0	0
29	5784	45	5829
3	11062	45	11107
4	18418	45	18463
2	20873	45	20918
7	24109	45	24154
5	26064	45	26109
11	36060	45	36105
13	57145	45	57190
12	57608	45	57653
15	64339	45	64384
16	68970	45	69015
17	72259	45	72304
1	123252	0	123252

Tabla 5: Scheduling Ruta 1

Scheduling Ruta 2:

Cliente	Tiempo Llegada	Tiempo de Servicio	Tiempo de Salida
1	0	0	0
26	15854	45	15899
28	25879	45	25924
27	29158	45	29203
6	35309	45	35354
35	53914	45	53959
18	62902	45	62947
19	66723	45	66768
36	69049	45	69094
34	69502	45	69547
14	80222	45	80267
20	89451	45	89496
1	154393	0	154393

Tabla 6: Scheduling Ruta 2

Scheduling de Ruta 3:

Cliente	Tiempo Llegada	Tiempo de Servicio	Tiempo de Salida
1	0	0	0
9	33286	45	33331
10	34713	45	34758
8	88004	45	88049
21	88917	45	88962
22	89415	45	89460
23	89947	45	89992
25	91112	45	91157
24	91649	45	91694
32	127483	45	127528
31	132668	45	132713
30	138950	45	138995
1	182329	0	182329

Tabla 7: Scheduling Ruta 3

Scheduling Ruta 4:

Cliente	Tiempo Llegada	Tiempo de Servicio	Tiempo de Salida
1	0	0	0
33	49427	45	49472
1	98899	0	98899

Tabla 8: Scheduling Ruta 4

6. Conclusiones y Recomendaciones.

Gracias a los modelos matemáticos e informáticos utilizados en este proyecto, se pueden concluir lo siguiente:

Dado el ineficiente sistema con el que la empresa trabaja actualmente, podemos concluir que al implementar un VRPTW, y la modelización presentada, existe una mejora promedio del 42% en el kilometraje recorrido lo cual también impacta de manera directa en el costo del flete y otros costos asociados, los mismos que se derivan en costos de combustible, mantenimiento, etc.

Como la distancia recorrida no es la misma todos los días, se puede generar un ahorro significativo al final del mes. Existiendo además, la posibilidad de minimizar la cantidad de personal (choferes) dado que existe una mejor administración en las rutas de las flotas de vehículos con las que actualmente trabaja la empresa. Sin embargo, este factor siempre va a depender de la demanda y el índice de rotación de los productos, entre otras variables que se deberían evaluar dentro de su gestión de inventarios.

Así mismo, el ahorro en kilometraje, podría ayudar a atender a más clientes, ya que se reduce el tiempo de entrega de los pedidos, y se podría cubrir algún requerimiento de emergencia por parte del cliente, pues en los negocios de cultivos, siempre se debe tratar que los clientes se sientan protegidos y respaldados por los servicios que la empresa estaría dispuesta a brindar.

Además, con los resultados obtenidos podemos concluir que este tipo de herramientas matemáticas, y de programación que combinan Metaheurísticas con algoritmos de mejora, son de gran ayuda para la correcta planificación de carga de las empresas, quienes buscan satisfacer a un alto nivel las necesidades de sus clientes, para crecer más en el mercado y por consiguiente ser más competitivos.

Por otro lado, no se puede negar que en nuestro medio existen muchos empresarios que se niegan a optar por este tipo de herramientas, esto sucede en la mayoría de

los casos por desconocimiento de las mismas o costumbre; es decir, por tener varios años manejando su negocio de manera empírica creyendo ser una buena alternativa para el mismo, lo cual los lleva a cometer muchos errores ocultos, sin saber que pueden mejorar y optimizar los procesos, porque este tipo de métodos son más efectivos, según lo demostrado para este caso en particular.

Para futuras investigaciones que se encuentren en este ámbito damos las siguientes recomendaciones:

- Una vez obtenidos los resultados y comparados los diferentes métodos para planificación de carga, podemos constatar que al utilizar metodologías apropiadas para abarcar este tipo de problemas permiten a las empresas mejorar eficaz y eficientemente; por consiguiente, se recomienda a la empresa de venta de agroquímicos invertir e implementar en este tipo de programas para así obtener resultados a mediano y largo plazo que repercutirán en ahorros significativos para su negocio.
- Utilizar herramientas como Visual Basic, para crear ventanas de ingreso de datos, conectadas a programas de optimización para generar rutas de manera más rápida.
- Implementar capacitaciones, acerca de temas de optimización, metaheurísticas, heurísticas, etc.
- Fomentar la promoción de las carreras dedicadas al estudio de estos procedimientos, abriendo las opciones de los empresarios ecuatorianos.
- Implementar algoritmos de mejora más eficaces, de manera que los resultados sean mejores, permitiendo el uso de la información por parte de los altos mandos de las empresas.
- Aceptar el uso de las herramientas informáticas y matemáticas en la optimización de los procesos operativos de las empresas.

7. Referencias Bibliográficas.

- [1] Delgado E., Modelos Avanzados en Transporte, Instituto de Ciencias Matemáticas, Escuela Superior Politécnica del Litoral, 2009.
- [2] Eidelman, A. y Valdez, A. Sistema de optimización para el ruteo dinámico de vehículos con ventanas de tiempo, Universidad de Buenos Aires Argentina, 2007.
- [3] Glover, F. et al. "Handbook of Metaheuristics", Kluwer Academic Publishers, Primera Edición, 2003, pp. XI.
- [4] Grosso de la Vega, R. Cálculo de Rutas de Vehículos de Reparto en una Ciudad con Ventanas Temporales de Acceso, Abril 2010.
- [5] Kwang Lee, Y., Mohamed, A., El-Sharkawi "Modern Heuristic Optimization Techniques", pag. 126.
- [6] Olivera, A. Heurísticas para Problemas de Ruteo de Vehículos, Universidad de la República, Montevideo – Uruguay, 2004.
- [7] Perla, en Judea. Heurística: Estrategias Inteligentes de la búsqueda para resolver problema informático, Nueva York, Addison-Wesley, 1983.
- [8] Sandoya F., Métodos Exactos y Heurísticos para resolver el Problema del Agente Viajero (TSP) y el Problema de Ruteo de Vehículos (VRP), Decimocuartas Jornadas en Estadística e Informática, Escuela Superior Politécnica del Litoral, Octubre 2007.
- [9] Wikipedia, "Algoritmo de Recocido Simulado", www.wikipedia.org.