



Escuela Superior Politécnica del Litoral

Facultad de Ciencias Naturales y Matemáticas

Ingeniería en Logística y Transporte

“Desarrollo de una metaheurística para resolver el problema de asignación
vehicular”

**INFORME DE MATERIA INTEGRADORA DE INGENIERÍA EN LOGÍSTICA Y
TRANSPORTE**

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

Presentado por:
Jonathan German Campoverde Salguero

Guayaquil – Ecuador

2016

AGRADECIMIENTOS

Quiero agradecer a Dios sobre todo por mantenerme y guiarme por los caminos correctos de la vida, ofreciéndome sabiduría y confianza para seguir adelante.

A mis padres el Ing. German Campoverde y la Dra. Gina Salguero que con buenos valores y consejos, ayudarán alcanzar mis objetivos personales y académicos, A mi abuelita y a mi hermano por ser pilares fundamentales en mi vida.

A mis maestros universitarios especialmente a los Ing. Xavier Cabezas, Víctor Vega, Johni Bustamante y Guillermo Baquerizo, quienes a través de sus cátedras y consejos contribuyeron a mi formación profesional.

Por ultimo a mis amigos del barrio, por su amistad incondicional.

Jonathan German Campoverde Salguero

DEDICATORIAS

Este trabajo lo dedico a toda mi familia y amigos, de manera especial a mis abuelos Efraín y Leónidas, mis tíos Cesar, Francisco y Jaqueline, personas que siempre estarán en mi corazón. A mis padres por darme ejemplo de responsabilidad y sencillez que a pesar de todo siguen adelante por demostrarme todo su amor y apoyo incondicional.

Jonathan German Campoverde Salguero

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE CIENCIAS NATURALES Y MATEMÁTICAS

INGENIERÍA EN LOGÍSTICA Y TRANSPORTE

INFORME DEL PROFESOR DE LA MATERIA INTEGRADORA

Habiendo sido nombrado PROFESOR DE LA MATERIA INTEGRADORA, del señor:

JONATHAN GERMAN
CAMPOVERDE SALGUERO

Con el tema del proyecto integrador *DESARROLLO DE UNA METAHEURÍSTICA PARA RESOLVER EL PROBLEMA DE ASIGNACIÓN VEHICULAR*, previa a la obtención del título de **INGENIERO EN LOGÍSTICA Y TRANSPORTE**, me permito informar que he leído el contenido y he revisado el formato del proyecto integrador, luego de lo cual indico que estoy de acuerdo en que el mismo se lo ha desarrollado conforme a los lineamientos de la Unidad de Titulación Especial de la ESPOL.

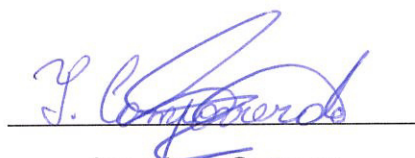
Guayaquil,


GUILLERMO BAQUERIZO PALMA
PROFESOR DE LA MATERIA INTEGRADORA

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este Informe del Proyecto de Graduación nos corresponden exclusivamente; y el patrimonio intelectual de la misma a la
ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL)



Jonathan German

Campoverde Salguero

RESUMEN

Este proyecto es realizado a una empresa de servicios que brinda el tratamiento de desechos contaminantes al medio ambiente (industriales, tóxicos, radioactivos, especiales) ubicada en Guayaquil, quien además gestiona su propia metodología de asignación de rutas de cobranza a sus clientes, generándose un problema ya que no cuentan con un sistema óptimo de planificación de rutas.

El objetivo principal del presente proyecto es realizar un sistema de generación de rutas que cubra los problemas que existen al visitar a los clientes y a su vez apoyándonos con técnicas metaheurísticas que permitan reducir la distancia total recorrida por los motorizados de la empresa.

Este problema de ruteo vehicular con ventanas de tiempo se lo resolverá aplicando la metodología GRASP para optimizar la distancia total recorrida.

Se tiene la posibilidad de trabajar con flota homogéneas de vehículos con costos fijos y variables diferenciados, velocidad, tiempos de duración de viajes y tiempo de servicio.

ABSTRACT

This project is realized to a service company that provides the treatment of polluting waste to environment (industrial, toxic, radioactive, special) located in Guayaquil, which also manages its own assignment methodology of collection routes to their customers, which means a problem because they do not have an optimal route planning system

The main objective of this project is to route generation system covering the problems that exist when visiting customers and in turn supporting us with metaheuristic techniques to reduce the total distance traveled by motor company.

This vehicular routing problem with time windows will be solved using the GRASP methodology to optimize the total distance traveled.

There is the possibility of working with homogeneous fleet of vehicles with different variables and fixed costs, speed, duration of travel time and service times

TABLA DE CONTENIDOS

AGRADECIMIENTOS	II
DEDICATORIAS	III
DECLARACIÓN EXPRESA	IV
RESUMEN.....	V
ABSTRACT.....	VI
INDICE DE GRÁFICOS	IX
INDICE DE TABLAS	X
GLOSARIO DE TÉRMINOS.....	XI
ABREVIATURAS.....	XII
CAPÍTULO 1	1
PLANTEAMIENTO DEL PROBLEMA	1
ANTECEDENTES	1
JUSTIFICACIÓN DEL PROBLEMA	2
HIPÓTESIS DE TRABAJO.....	3
OBJETIVO GENERAL.....	3
OBJETIVOS ESPECÍFICOS.....	3

CAPÍTULO 2	4
MARCO TEÓRICO	4
REVISIÓN DE LA LITERATURA	4
MARCO CONCEPTUAL	8
Definiciones básicas	8
Formulación Matemática.....	11
Técnicas de Resolución de Problemas de Optimización Combinatoria	14
CAPÍTULO 3	18
METODOLOGÍA DE TRABAJO	18
Cronograma de Actividades	20
ORGANIGRAMA DE LA EMPRESA	21
Funciones de los diferentes cargos de la empresa <i>ABC</i>	22
Capítulo 4	27
Planteamiento de la solución.....	27
Clasificación de clientes por zona	27
Ubicación Geográfica de los clientes	28
Aplicación de la metodología GRASP	28
Análisis y comparación de resultados	37
Bibliografía	38

INDICE DE GRÁFICOS

Gráfico 1. 1 Situación actual de la empresa.....	1
Gráfico 1. 2 Problemática	2
Gráfico 3. 1 Variantes del VRP	9
Gráfico 3. 2 Diagrama de flujo sobre la metodología	19
Gráfico 3. 3 Cronograma de actividades	20
Gráfico 3. 4 Organigrama de la empresa ABC	21
Gráfico 4. 1 Ubicación geo-referencial de los clientes	28
Gráfico 4. 2 Rutas resultantes GRASP	34
Gráfico 4. 3 Rutas fase de Mejora	36

INDICE DE TABLAS

Tabla 4. 1 Datos para la Metaheurística.....	30
Tabla 4. 2 Coordenadas Rectangulares.....	31
Tabla 4. 3 Resultados GRASP	33
Tabla 4. 4 Resultados Fase de mejora 2-OPT	35
Tabla 4. 5 Comparación entre F. Constructiva vs F. Mejora	37
Tabla 4. 6 Distancias Recorridas	37

GLOSARIO DE TÉRMINOS

Algoritmo	Procedimiento de cálculo con símbolos, según unas reglas determinadas y con un número finito de pasos.
Determinado	Según la alternativa previamente acordada.
Distancia de Manhattan	Es la distancia entre dos puntos, medida sobre ejes a ángulos rectos, es decir, la distancia que se recorrería para llegar de un punto a otro si se siguiera una trayectoria en forma de cuadrícula, por ejemplo, ir en auto por una ciudad con estructuras distribuidas en cuadras.
Empírico	Sistema o procedimiento fundado en una práctica o rutina
Homogéneo	Relativo a un mismo género.
Heurísticas	Son procedimientos simples que realizan una exploración limitada del espacio de búsqueda y dan soluciones de calidad aceptable en tiempos de cálculo generalmente moderados.
Inserción	Lugar donde una cosa se inserta en otra
Optimizar	Buscar la mejor manera de realizar una actividad.
Optimización Combinatoria	Es una rama de la optimización en matemáticas aplicadas y en ciencias de la computación, relacionada a la investigación de operaciones, teoría de algoritmos y teoría de la complejidad computacional.
Parámetro	Variable que toma un valor constante durante la ejecución de un programa.
Recorrido	Conjunto de todos los valores alcanzados por una función.
Traveling Salesman Problem (TSP)	Problema de optimización donde se dispone de un solo vehículo que debe visitar a todos los clientes en una sola ruta y a costo mínimo. No suele haber un depósito (y si lo hubiera, no se distinguiría de los clientes), no hay demanda asociada a los clientes y tampoco hay restricciones temporales.
2-OPT	Es un algoritmo de búsqueda local, fue propuesto por primera vez en 1958 para resolver el problema del agente viajero (TSP). La idea principal detrás de esto es tomar una ruta que cruza sobre sí mismo y reordenarla de tal manera que la solución inicial sea mejorada.

ABREVIATURAS

GRASP	Greedy Randomized Adaptive Search Procedures (Procedimiento adaptivo de búsqueda aleatorizada voraz)
GREDDY	Algoritmo Voraz
MDVRP	Multiple Depot Vehicle Routing Problem (Problema de ruteo vehicular con depósitos múltiples)
PVRP	Periodic Vehicle Routing Problem (Problema de ruteo vehicular periódico)
RL	Restricted List (Lista Restringida)
SL	Solution List (Lista de solución)
TSP	Traveling Saleman Problem (Problema del Agente Viajero)
VRP	Vehicle Routing Problem (Problema de ruteo vehicular)
VRPPD	Vehicle Routing Problem with Pickup and Delivery (Problema de ruteo vehicular con recolección y entrega)
VRPTW	Vehicle Routing Problem with Time Window (Problema de ruteo vehicular capacitado con ventanas de tiempo)

CAPÍTULO 1

PLANTEAMIENTO DEL PROBLEMA

ANTECEDENTES

La empresa objeto de estudio se dedica a brindar soluciones ambientales y de salud en cuanto a la gestión integral de los residuos peligrosos y de alto grado de contaminación. Esta empresa empezó en el año 2003, en Guayaquil. Además, desarrollan actividades de Recolección, Transporte, Almacenamiento, Capacitación, Tratamiento, y disposición final de residuos especiales y peligrosos.

En la actualidad, debido a la situación económica que atraviesa el país, la empresa mantiene en una constante actualización en sus procesos logísticos, para reducir costos de operación en todas sus áreas, por esta razón se ven en la obligación de implementar rutas de recolección de documentos financieros, el cual pueda ser llevado a la práctica como un método eficiente y satisfactorio para las necesidades de la empresa, como se muestra en el gráfico 1.1

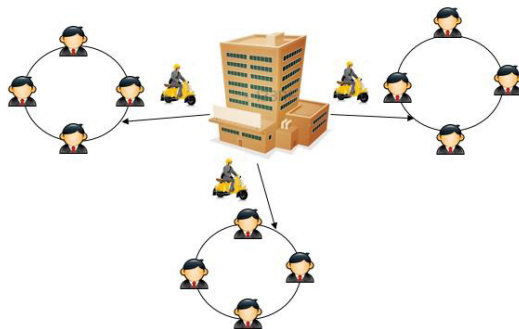


Gráfico 1. 1 Situación actual de la empresa

Fuente: Creada por el autor

JUSTIFICACIÓN DEL PROBLEMA

La empresa al no tener una buena planificación de rutas de cobranza, no cumple satisfactoriamente con las metas diarias establecidas, por lo tanto, desean saber qué cantidad de mensajeros motorizados se requiere para la operación de recolección y transportación de documentos financieros.

Este problema conlleva a que el flujo de dinero que entra a la empresa no sea el óptimo ya que se encuentran falencias al momento de realizar la operación, éstas pueden ser por:

- Retrasos al momento de ir a cobrar a los clientes.
- Falta de coordinación en la visita a los clientes (cruce de motorizados en la misma zona).
- Problemas en la gestión de los documentos financieros.

En el gráfico 1.2 se muestra la problemática que actualmente se encuentra en el área de cobranzas, por lo que no utilizan una buena planificación al momento de diseñar las rutas, por esta razón los motorizados se encuentran un tanto desorientados al momento de realizar la operación de cobranzas.



Gráfico 1. 2 Problemática

Fuente: Creada por el autor

HIPÓTESIS DE TRABAJO

Elaborar una óptima planificación de rutas permitirá cumplir las metas establecidas por el departamento de cobranzas y que el ingreso de flujo de efectivo sea más rentable así se reducirán costos de movilización y los tiempos de visita a los clientes.

OBJETIVO GENERAL

Diseñar rutas óptimas para la recolección de documentos financieros

OBJETIVOS ESPECÍFICOS

- Desarrollar rutas para un determinado número de clientes
- Mejorar la eficiencia de la operación
- Cubrir los tiempos de visitar a los clientes

CAPÍTULO 2

MARCO TEÓRICO

En este capítulo se mencionará algunas metodologías que facilitarán el procedimiento de resolver el problema de ruteo vehicular con ventanas de tiempo (VPTW), las cuales nos servirán para garantizar el cumplimiento de los objetivos planteados anteriormente, además se mencionarán conceptos relevantes al tema.

Es importante diseñar buenas rutas y cumplir los tiempos de visitas a los clientes para mejorar el nivel de servicio brindado por parte de la empresa.

REVISIÓN DE LA LITERATURA

En esta sección se revisará como objetivo principal las teorías, ideas y aplicaciones acerca del problema de ruteo vehicular citando algunas aplicaciones realizadas por autores cuyas investigaciones ayudaron a la realización del presente proyecto. Por lo cual se analizó los informes científicos realizados por ellos, entre estos constan:

Un algoritmo genético híbrido generacional para el problema periódico de rutas para vehículos con ventanas de tiempo, Montréal, Canada.

En este trabajo realizado por los Señores P.K. Nguyen, T.G. Crainic y M. Toulouse en el año 2014 se describe el desarrollo de una metaheurística híbrida basada en una población para el ruteo de vehículos con ventanas de tiempo, esta metaheurística es un algoritmo genético generacional que utiliza dos metaheurísticas basadas en el vecindario para optimizar la descendencia, las cuales se utilizan por su capacidad para escapar de los óptimos locales y ofrecer soluciones optimizadas y diversificadas para la población de la próxima generación.

El algoritmo genético que proponen estos señores presenta dos nuevos cruces de operadores que abordan el problema de rutas para vehículos con ventanas de tiempo, los dos operadores que se cruzan buscan la diversificación de la exploración en el espacio de la solución, preservando al mismo tiempo la información de las rutas de la población, así como las rutas de optimización NP-duro, nos demuestran que es una metodología altamente competitiva, proporcionando nuevas soluciones factibles.

Aplicación de la metodología GRASP al problema de Ruteo de Vehículos

El estudio realizado por los Sres. Paolo Priore, Raúl Pino, Carlos Martínez, Verónica Villanueva e Isabel Fernández en el año 2013, se describe el desarrollo e implementación de un Sistema de Soporte a la Decisión (DSS) que ayudará en el proceso de cálculo de rutas en la ciudad de Gijón-España.

Principalmente se analizó el comportamiento de las distintas metodologías heurísticas existentes para poder resolver un problema de rutas del tipo MDVRP y VRPTW. El autor adoptó la metodología GRASP como núcleo optimizador en el momento de la generación de rutas

Las soluciones obtenidas dan evidencia de la mejora encontrada entre estos resultados y los anteriores, ya que se creaban rutas con mayor demanda dejando en espera envíos a clientes con poca demanda hasta poder completar un vehículo, lo que provocaba retrasos en las fechas programadas.

El algoritmo trata de encontrar combinaciones entre todos los clientes asegurando así que el número máximo de vehículos sean enviados sin excederse de la fecha programada, demostrando así la eficacia de la aplicación de la metodología.

GRASP mejorado con búsqueda tabú para el problema de enrutamiento tanto con ventana de tiempo y número limitado de vehículos, Hong Kong, China.

En este trabajo realizado en el 2014 por Zhiye Li, Songshan Guo, Fan Wang y Andrew Lim para la Universidad de Hong Kong-China, se analiza una nueva extensión usable para el problema de ruteo vehicular tanto con ventanas de tiempos y número limitado de vehículos.

Se estudia con un mejor procedimiento de búsqueda aleatorizado Greedy mediante técnicas de múltiples inicializaciones, reutilizando la solución y la mejora de la mutación, con cuatro heurísticas específicas para VRPTW; en primer lugar se utiliza el menor tiempo, en segundo lugar se basa en los clientes que están más cercanos, y por ultimo continuaremos con los de menos tiempo de espera.

A partir de los resultados experimentales sobre los datos de referencia, se demuestra que con nuestro algoritmo se soluciona el problema VRPTW con un rendimiento estable y en un tiempo de ejecución corto.

De hecho, las técnicas de búsqueda de propuestas en este documento se pueden aplicar fácilmente para otros meta-heurísticos para la resolución de problemas de este tipo.

Esta investigación es fundamental para el desarrollo de este proyecto debido a que el enfoque es similar, el cual es de gran importancia en el ámbito logístico para optimizar recursos.

MARCO CONCEPTUAL

En esta sección se analizará algunos conceptos, teorías, definiciones y demás temas actualizados que servirán para la elaboración y análisis de este proyecto.

Definiciones básicas

- **Optimización**

Este término se lo utiliza cuando queremos encontrar la mejor solución dentro de un conjunto de posibles soluciones generadas al resolver un modelo matemático, el cual es un problema de optimización combinatoria que al ser resuelto genera un conjunto de soluciones posibles y diferentes cuyos objetivos fundamentales es encontrar cuál de estas me generan la mejor solución.

- **Problema de Ruteo Vehicular**

Dantzig y Ramser en el año 1959, introdujeron por primera vez el problema de ruteo vehicular, quienes al observar un problema en la entrega de combustible en una estación de servicio lo tomaron como una aplicación real, luego de casi cinco años después en 1964, Clarke & Wright elaboraron y propusieron la primera heurística efectiva para la resolución del VRP. De esta manera abren camino las investigaciones para esta área específica.

El VRP busca determinar un conjunto de rutas optimas a un costo mínimo de enrutamiento que inicien y finalicen en los depósitos, teniendo en cuenta que tenemos un conjunto de clientes y depósitos ubicados en distintos puntos geográficos y una flota determinada de vehículos, teniendo como restricción básica la única visita a cada cliente en cualquier vehículo de la flota.

- **Variantes del VRP**

El problema de ruteo vehicular presenta algunas variantes que se detallaran a continuación:

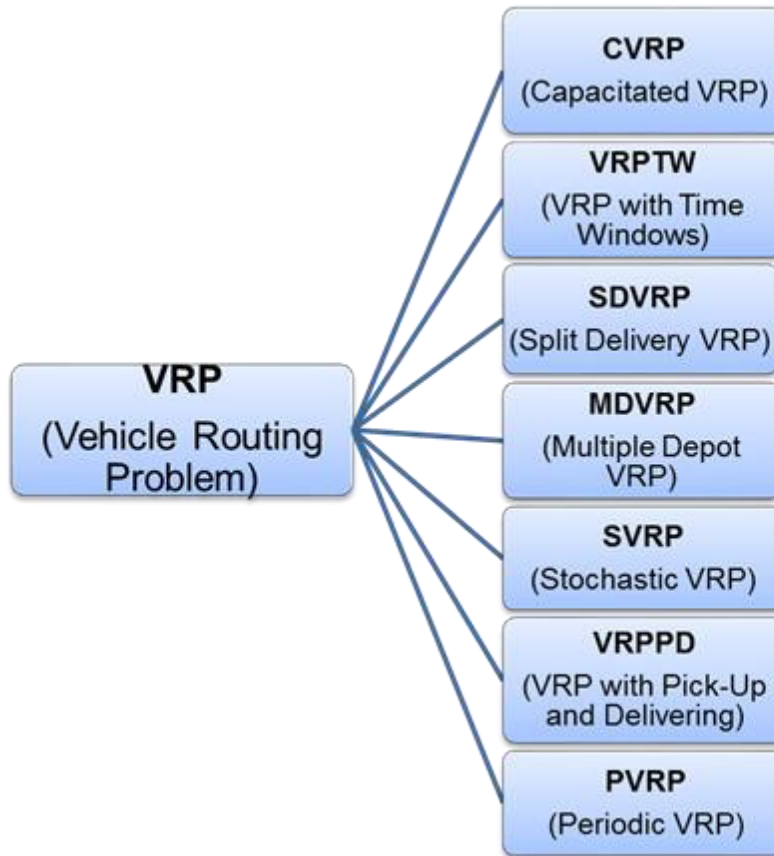


Gráfico 3. 1 Variantes del VRP

Fuente: William J. Cook

- **CVRP:** Conocido como VRP capacitado, tiene una capacidad para cada vehículo.
- **VRPTW:** Conocido como VRP con ventanas de tiempo, en donde los clientes son atendidos en un cierto espacio de tiempo, los cuales se las llaman ventanas de tiempos, y estas pueden ser duras o suaves, la

diferencia de estas radica en el momento de la formulación, ya que una ventana dura debe ser puesta como una restricción del problema, a diferencia de una ventana suave que puede ser tomada como una penalización a la función objetivo en caso de incumplirla.

- **SDVRP:** La entrega de la mercadería a un cliente puede ser atendida en más de un vehículo.
- **MDVRP:** VRP con múltiples depósitos.
- **SVRP:** Las demandas, los clientes y los tiempos de servicios pueden ser aleatorios.
- **VRPPD:** Se atienden a todos los clientes de una ruta, entregando y recolectando simultáneamente.
- **PVRP:** El proveedor y el cliente pactan la entrega de la mercadería d forma periódica.

El presente trabajo se enfoca en la variante VRPTW debido a que la empresa objeto de estudio cuenta con una flota homogénea y los clientes deben ser atendidos en ocasiones dentro de espacios de tiempo establecidos.

- **VRPTW (Vehicle Routing Problem-Time Windows)**

En este problema se debe encontrar la ruta óptima teniendo en cuenta que cada cliente tiene una ventana de tiempo (Inicio-Fin), y los vehículos deben atender a los clientes en estas horas fijas pactadas anteriormente, por lo cual existen donde tipos de ventanas de tiempo:

- Duras: Obligatoriamente tienen que ser atendidos los clientes en este espacio de tiempo.
- Suaves: Existe un rango de tiempo establecida por el cliente en el cual debería ser atendido y una función de costo para retrasos o adelantos (penalización en la función objetivo).

Formulación Matemática

La formulación matemática del VRPTW, es el clásico modelo del VRP pero esta vez se añade las restricciones de las ventanas de tiempo:

Dado un grafo dirigido $G = (V, C)$ donde V es el conjunto de vehículos de flota homogénea, C el conjunto de clientes:

$$V = \{v_0, v_1, v_2, \dots, v_n\} \quad C = \{(v_i, v_j) : v_i, v_j \in V, v_i \neq v_j\}$$

El VRPTW consiste en la elaboración de un conjunto determinadas d rutas que satisfagan las siguientes restricciones:

- Cada ruta comienza y termina en el depósito.
- Cada cliente es visitado una y solo una vez por un solo vehículo.
- La demanda de cada cliente se satisface en cada visita.
- Los clientes son atendidos dentro de las ventanas de tiempo.
- La demanda total no excede la capacidad de cada vehículo.

Variables:

A continuación, se presenta los parámetros y variables del VRPTW:

- n : Número de clientes donde el nodo 0 representa al depósito.
- M : Numero de vehículos usados.
- V : Conjunto de n clientes.
- E : Conjunto de los posibles arcos
- S : Subconjunto del conjunto de clientes V .
- $Arco(i,j)$: Representa el mejor camino para ir desde el nodo i hacia el nodo j en la red de transporte.
- $C(i,j)$: Costo asociado al $Arco(i,j)$ en este proyecto el grafo con el que se trabajara es simétrico es decir el costo de ir de i a j es exactamente el mismo que ir de j a i .

Se puede definir la siguiente variable binaria:

$$X_{ijm} \begin{cases} 1, \text{ si el arco } (v_i, v_j) \in C \text{ es considerado en la ruta de un vehiculo } k \in K \\ 0, \text{ caso contrario} \end{cases}$$

La Función objetivo consiste en sumar todos los costos asociados a los arcos que pertenecen a una solución dada:

$$Min Z = \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk}$$

Restricciones:

- Asegura que cada cliente sea visitado una sola vez por un vehículo:

$$\sum_{k \in V} \sum_{j \in N} x_{ijk} = 1 \quad \forall i \in C$$

- Esta restricción permite que cualquier vehículo se carga, como mucho, a su capacidad.

$$\sum_{i \in C} q_i \sum_{j \in N} x_{ijk} \leq Q \quad \forall k \in V$$

- Las siguientes tres restricciones son ecuaciones de flujo que garantizan que el vehículo sale del nodo 0 una sola vez, que abandona cualquier nodo i , si y solo si ha entrado antes en él, y al final vuelve al nodo $n+1$.

$$\sum_{j \in N} x_{0jk} = 1 \quad \forall k \in V$$

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0 \quad \forall h \in C, \forall k \in V$$

$$\sum_{i \in N} x_{i,n+1,k} = 1 \quad \forall k \in V$$

- Esta restricción establece que el vehículo k no puede llegar a j antes de $b(i,j) + t(i,j)$, cuando se viaja de i a j .

$$x_{ijk}(b_{ik} + t_{ij} - b_{jk}) \leq 0 \quad \forall i, j \in N, \forall k \in V$$

- Estas últimas garantizan que se respeten todas las ventanas de tiempo temporales y que la variable $X(i,j)$ sea entera.

$$e_i \leq b_{ik} \leq u_i \quad \forall i \in N, \forall k \in V$$

$$x_{ijk} \in \{0,1\} \quad \forall i,j \in N, \forall k \in V$$

Es importante recordar que el problema de ruteo vehicular con ventanas de tiempo aplicado a este proyecto tiene como demanda del cliente unitario por el tanto la capacidad de los vehículos que en este caso con motorizados también sea unitaria, pero se podría considerar la condición de máxima capacidad de transportar dinero o documentos.

TÉCNICAS DE RESOLUCIÓN DE PROBLEMAS DE OPTIMIZACIÓN

COMBINATORIA

- **Metaheurística GRASP**

Los métodos de GRASP fueron desarrollados al final de la década de los 80 con el objetivo inicial de resolver problemas de cubrimientos de conjuntos (Feo y Resende, 1989). El término GRASP fue introducido por Feo y Resende (1995) como una nueva técnica metaheurística de propósito general.

GRASP es un procedimiento de multi-arranque en donde cada paso consiste en una fase de construcción y una de mejora. En la fase de construcción se aplica un procedimiento heurístico constructivo para obtener una buena solución inicial. Esta solución se mejora en la segunda fase mediante un algoritmo de búsqueda local. La mejor de todas las soluciones examinadas se guarda como resultado final.

En la fase de construcción se construye iterativamente una solución posible, considerando un elemento en cada paso. En cada iteración la elección del próximo elemento para ser añadido a la solución parcial viene determinada por una función greedy. Esta función mide el beneficio de añadir cada uno de los elementos según la función objetivo y elegir la mejor. Notar que esta medida es miope en el sentido que no tiene en cuenta qué ocurrirá en iteraciones sucesivas al realizar una elección, sino únicamente en esta iteración.

Se dice que el heurístico greedy se adapta porque en cada iteración se actualizan los beneficios obtenidos al añadir el elemento seleccionado a la solución parcial. Es decir, la evaluación que se tenga de añadir un determinado elemento a la solución en la iteración i , no coincidirá necesariamente con la que se tenga en la iteración $i+1$.

El heurístico es aleatorizado porque no selecciona el mejor candidato según la función greedy adaptada, sino que, con el objeto de diversificar y no repetir soluciones en dos construcciones diferentes, se construye una lista con los mejores candidatos de entre los que se toma uno al azar.

Al igual que ocurre en muchos métodos, las soluciones generadas por la fase de construcción de GRASP no suelen ser óptimos locales. Dado que la fase inicial no garantiza la optimalidad local respecto a la estructura de entorno en la que se esté trabajando (notar que hay selecciones aleatorias), se aplica un procedimiento de búsqueda local como Post-procesamiento para mejorar la solución obtenida

En la fase de mejora se suele emplear un procedimiento de intercambio simple con el objeto de no emplear mucho tiempo en esta mejora. Notar que GRASP se basa en realizar múltiples iteraciones y quedarse con la mejor, por lo que no es especialmente beneficioso para el método el detenerse demasiado en mejorar una solución dada.

Además uno de los pasos más importantes al momento de desarrollar un procedimiento GRASP es tener claro cómo se va a graduar correctamente el parámetro α , teniendo en cuenta que puede estar entre 0 y 1 ($0 \leq \alpha \leq 1$), si se aproxima a 1 vamos a tener un procedimiento casi completamente glotón y si se aproxima a 0 se convierte en un algoritmo casi en su totalidad aleatorizado, también hay que tener en cuenta que este parámetro nunca debe de tomar ni el valor de 1 o 0 porque solo sería un greedy o un algoritmo aleatorizado.

Algoritmo

Mientras (Condición de parada)

Fase Constructiva

- Seleccionar una lista de elementos candidatos. (CL), luego se procede a la evaluación de cada candidato en la función greedy, de esta manera se puede saber que aportará a la función objetivo cada uno de ellos.
- Considerar una Lista Restringida de los mejores Candidatos. (RCL), en este parte se hace uso de intervalos de aceptación que se obtiene gracias al parámetro α .

- Seleccionar un elemento aleatoriamente de la Lista Restringida.
- Este procedimiento se repite hasta tener completa la solución inicial, en el caso de un CVRP hasta que la lista de candidatos CL no tenga elementos, y todos formen parte de la solución.

Fase de Mejora

Realizar un proceso de búsqueda local a partir de la solución construida hasta que no se pueda mejorar más, en este caso se usa un procedimiento 2-opt como método de búsqueda local.¹

Actualización

Si la solución obtenida mejora a la mejor almacenada, actualizarla.

CAPÍTULO 3

METODOLOGÍA DE TRABAJO

Se establecerá un método capaz de realizar el trabajo, esto quiere decir que la recolección, el análisis y la evaluación de la información para el desarrollo de este sean satisfactorios.

Recolectada la información de la empresa objeto del presente estudio, se procederá a desarrollar la solución para el problema propuesto en este trabajo.

Mediante la colaboración del personal que labora en la empresa ABC, se logró obtener información como:

- ✓ Direcciones de los clientes
- ✓ Cantidad de clientes visitados diariamente
- ✓ Costos de movilización diarios en visitar a un cliente.

La metodología a implementarse dentro del presente proyecto se detalla a continuación:

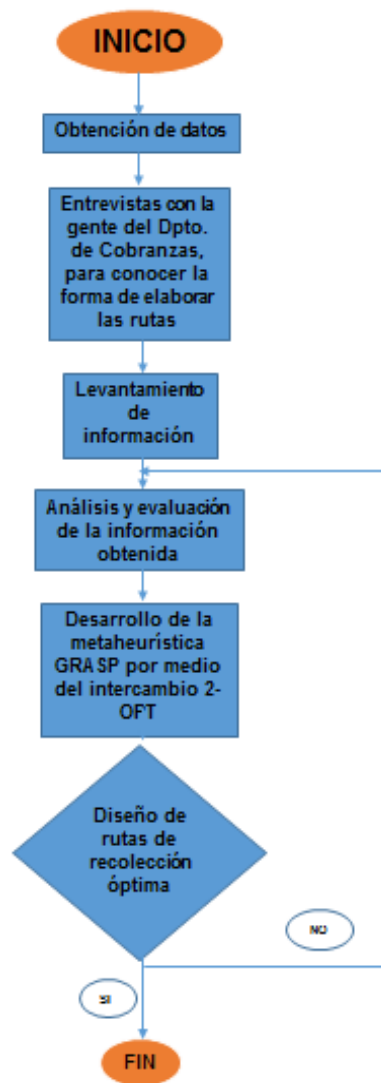


Gráfico 3. 2 Diagrama de flujo sobre la metodología

Fuente: Creada por el autor

CRONOGRAMA DE ACTIVIDADES

Las actividades a realizarse dentro del presente estudio se detallan a continuación con sus periodos y tiempos en que cada una de ellas se ejecutará:

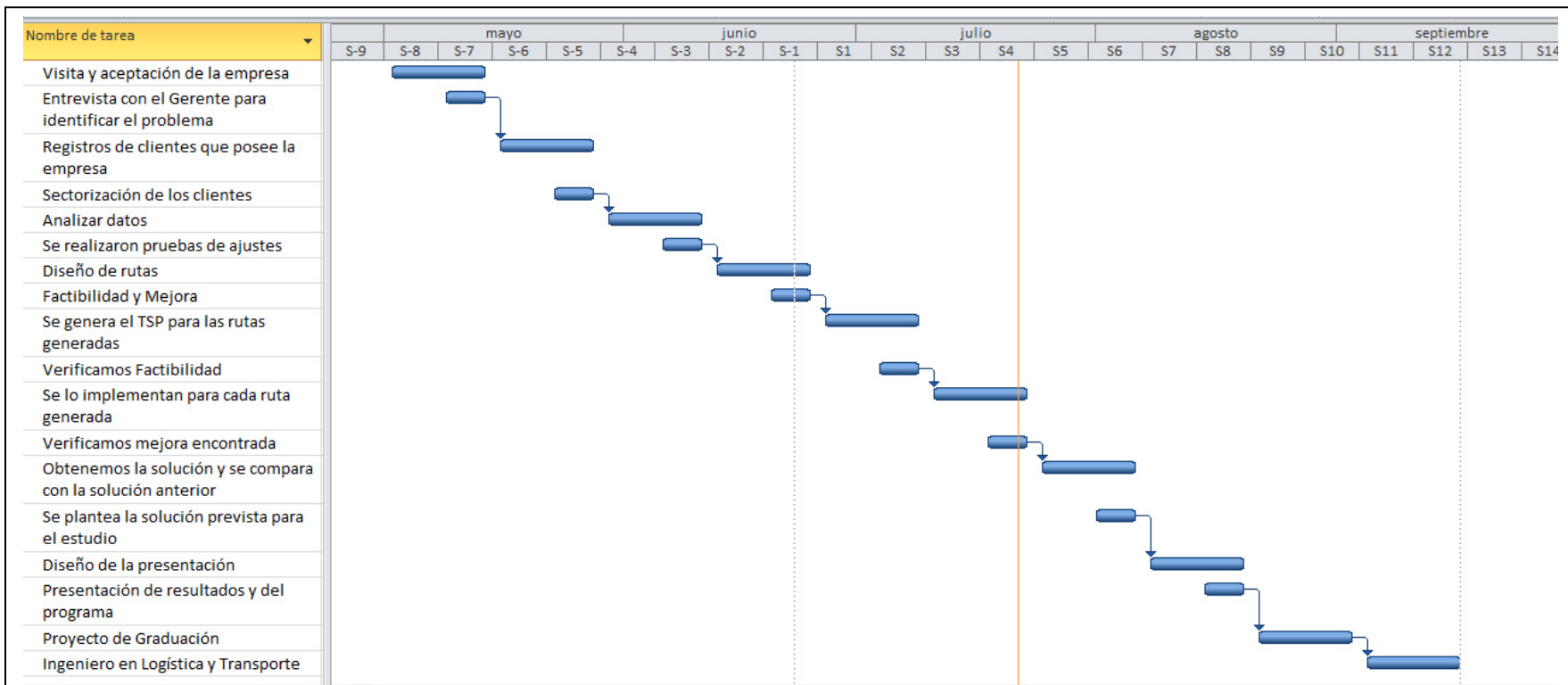


Gráfico 3. 3 Cronograma de actividades

Fuente: Creada por el autor

Organigrama de la empresa

ORGANIGRAMA DE LA EMPRESA

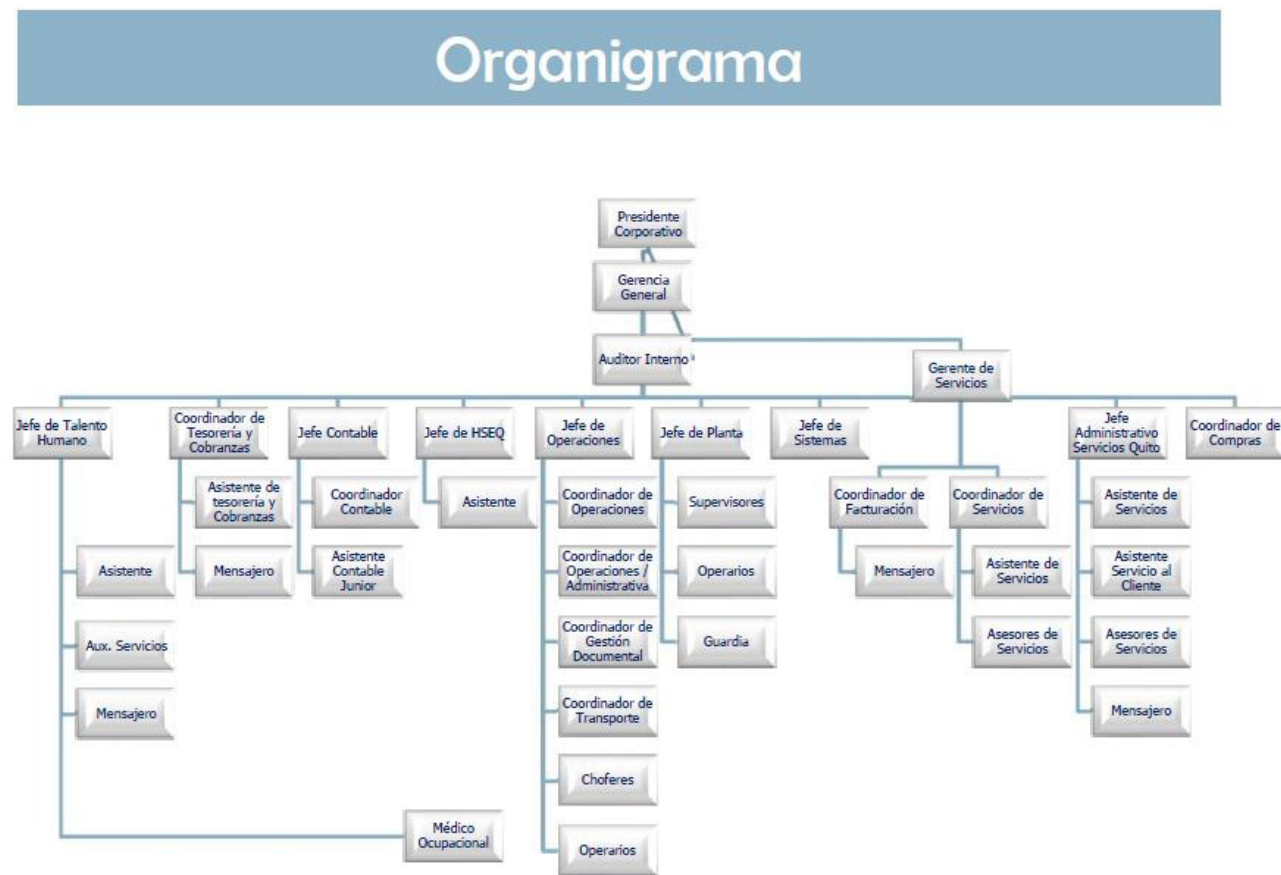


Gráfico 3. 4 Organigrama de la empresa ABC

Fuente: Empresa ABC

FUNCIONES DE LOS DIFERENTES CARGOS DE LA EMPRESA ABC

Gerente General

- Plantear el proceso estratégico de la organización, estableciendo los objetivos y metas específicas de la empresa.
- Desarrollar estrategias generales para alcanzar los objetivos y metas propuestas.
- Fomentar la integración de los trabajadores para que puedan lograr las metas tanto individuales como grupales optimizando los recursos disponibles, como lo son el tiempo, dinero y materiales.
- Gestionar una estructura sólida administrativa que contenga los elementos necesarios para el desarrollo de los planes de acción laborables.

Talento Humano

- Supervisar las aptitudes de los trabajadores y que sigan las normas laborables de la empresa.
- Capacitar y brindar por medio de todas las vías de comunicación información necesaria a los trabajadores para obtener una mejora continua en sus áreas laborables.
- Establecer relaciones de beneficio mutuo con las entidades o clientes a fines o partícipes de los logros u objetivos planteados anteriormente.

Tesorería y Cobranzas

- Planificar, organizar y coordinar el área contable, con el objetivo de obtener las consolidaciones y estados financieros requeridos por la organización.
- Establecer y coordinar la ejecución de las políticas relacionadas con el área contable, asegurándose que se cumplan los principios de contabilidad generalmente aceptados y con las políticas específicas de la empresa.
- Asignar estos fondos en los proyectos de inversión más seguros y rentables.
- Determinar la política de dividendos más conveniente.

Facturación

- Planificar financieramente a corto y a largo plazo.
- Revisar y comparar lista de pagos, comprobantes, cheques y otros registros con las cuentas respectivas.
- Recibir, examinar, clasificar, codificar y efectuar el registro contable de documentos.
- Archivar documentos contables para uso y control interno.
- Elaborar y verificar relaciones de gastos e ingresos.
- Obtener retroalimentación del personal para obtener mejoras en los procesos.

Jefe de Seguridad Industrial (HSEQ)

- Controlar y administrar todos los procesos de seguridad de la empresa.
- Establecer políticas de seguridad industrial para todos los colaboradores de la empresa.
- Dar charlas de seguridad ambiental tanto para el cliente interno como el cliente externo.
- Examinar a diario las instalaciones de la empresa donde el índice de contaminación sea más alto.

Jefe de Operaciones

- Administra toda la cadena de suministro: recolección, almacenamiento, procesamiento.
- Controla la flota de transporte y las funciones que deriva esta.
- Asigna y distribuye las rutas de recolección de desechos.
- Asegurar que las actividades de su área de competencia se lleven a cabo respetando lo establecido en los sistemas de Calidad, Seguridad, Salud Ocupacional y medio Ambiente de la empresa.

Jefe de Planta

- Controla todo el proceso de la planta: incineración, tratamiento, y esterilización de los desechos recolectados a los clientes.

- Supervisa las funciones de los operarios.

Jefe de Sistemas

- Controla todos los sistemas informáticos que posee la empresa.
- Mantiene activo y funcional las computadoras de la empresa.
- Administra las licencias de los diferentes ERP que maneja la empresa.

Gerente de Servicios

- Administra el área de facturación de la empresa.
- Controla los mensajeros para los servicios que brinda la empresa.
- Brindar soluciones que estén dentro del paquete de servicios que brinda la empresa.

Jefe Administrativo servicios Quito

- Brinda los mismos servicios que se ofrecen en la matriz Guayaquil en Quito y en la región norte del país.
- Gestiona las cobranzas de la región
- Mantiene una ventanilla de servicio al cliente para atender cualquier novedad del cliente de la zona.

Coordinador de compras

- Gestiona todas las compras locales e internacionales que se requiere en la empresa.
- Satisface todas las necesidades adquisitivas que requiere cada departamento de la empresa.
- Proporciona asistencia en cuanto al área contable para administrar bien el presupuesto de la empresa.

CAPÍTULO 4

PLANTEAMIENTO DE LA SOLUCIÓN

En este capítulo se procederá a encontrar una solución para los problemas anteriormente planteados, para ello, será de gran utilidad un software de optimización matemática. En el medio computacional se encuentra una gran variedad de software de programación algebraica algunos de ellos son: GAMS, MATLAB, TORA, PHYTON, pero el más factible para la elaboración de este proyecto de graduación es “MATHEMATICA 10.1”, este software brinda todas las facilidades debido a que las posibles soluciones generadas crezcan de manera exponencial para ser resueltos en un tiempo prudente, otra de las facilidades que brinda este programador es que se puede interactuar en la programación mejorando su aspecto y presentación de los resultados más eficientes

La Metaheurística GRASP es un método de dos fases, la primera fase es la etapa constructiva donde se genera una solución inicial, posteriormente se continúa con la mejora local repitiéndose estas dos fases, almacenando la mejor solución encontrada en iteraciones anteriores.

CLASIFICACIÓN DE CLIENTES POR ZONA

Los clientes atendidos por la matriz son todos aquellos que pertenecen a la ciudad de Guayaquil, para el presente proyecto se ha elegido reunir aquellos que pertenezcan al Norte, Sur, Este, Oeste y Centro de la ciudad, el región norte esta subdividida por cuatro partes (norte 1, norte 2, norte 3, norte 4), el cual se escogerá aleatoriamente 10 clientes por cada subregión del sector norte, esto

dará una muestra de 40 clientes con una población total de 1300 clientes aproximadamente.

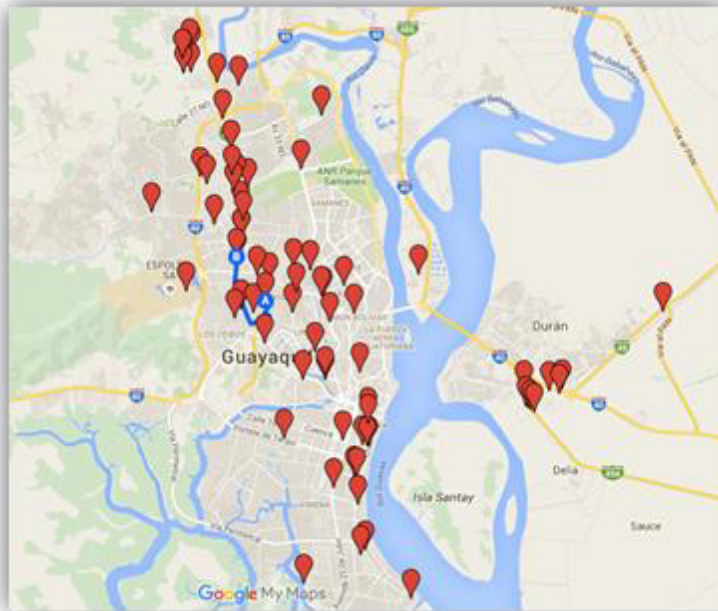


Gráfico 4. 1 Ubicación geo-referencial de los clientes

Fuente: Creada por el autor

UBICACIÓN GEOGRÁFICA DE LOS CLIENTES

Una vez establecido la cantidad de clientes que integran la zona a rutear, se procedió con la obtención de sus coordenadas geográficas mediante la ayuda de GOOGLE MAPS, localizando así cada punto con su respectiva longitud y latitud, según las direcciones de los clientes brindados por la empresa.

APLICACIÓN DE LA METODOLOGÍA GRASP

Para resolver el problema vehicular con ventanas de tiempo VRPTW se etiquetará a los clientes con una nomenclatura que facilitará el ingreso de datos a la aplicación antes mencionada. A continuación en la tabla se muestra los datos que serán usados para la heurística como el tiempo en trasladarse y dirección de cada

cliente, aparte de la ubicación en longitud y latitud que se la manejará con el software MATHEMATICA 10.

Direcciones	Minutos	tiempo de atención
URDENOR 1 MZ 115 V 8 (DETRAS DE CLINICA URDENOR)	8	17
SAUCES 3 MZ. 149 VILLA 17	4	29
ALBORADA SEXTA ETAPA SOLAR 26	6	11
CDLA. LAS ORQUIDEAS MZ 1017 V 20	10	26
ALBORADA 8VA ETAPA MZ 840 S 10-2	9	17
ALBORADA 4TA ETAPA AV. JOSE MARIA EGAS HERRADURA 45 MZ. DP VILLA 1	7	14
GUAYACANES MZ 100 VILLA 2	12	10
COLINAS DEL A ALBORADA SL 33 MZ 741	11	10
SAUCES IV MZ. 373F V. 50B	7	18
GARZOTA SOLAR 11 1ER PISO OFICINA 2	2	26
AV. JUAN TANCA MARENGO KM 4.5	18	17
KM 4.5 VIA DAULE	18	12
AV. JUAN TANCA MARENGO KM 2 ½	17	14
VIA A DAULE KM 7.5 PRIMERA 507 Y CALLE 11 AVA. - CALLE 12AVA (POR LA CANCHA SINTETICA)	25	25
KM.14.5 VIA A DAULE DIAG. MABE COOP.5 DICIEMBRE MZ 653 SOLAR 1	30	19
CIUDEDELA MARTHA DE ROLDOS MZ 607 V 18	16	21
ABEL CASTILLO ROMERO Y JUAN TANCA MARENGO TORRE MEDICA 2 OMNIHOSPITAL OF 606	15	28
CDLA. SAN FELIPE MZ. 162 VILLA 26	17	26
KM.12.5 VIA DAULE	28	25
CDLA. HUANCIVILCA MZ A-19 V. 4	21	18
MAPASINGUE OESTE CDLA. DEL BANCO DE LA VIVIENDA MZ. E V. 4	34	21
CDLA. FLORIDA NORTE MZ 116 SL 3	41	11
COOP. BASTION POPULAR MZ. 577 SL. 1	38	30
COOP. QUISQUIS SL. 4 A MZ. U-2	41	11
COOP PANCHO JACOME VIA DAULE SL 8 Y MZ 273 (ATRAS DE LA CASA COMUNAL).	36	17
CDLA FLORIDA NORTE MZ. 616 SL. 3	36	13
CDLA. QUISQUIS HECTOR TOSCANO # 12 y EUGENIO ESPEJO	44	12
KM. 26.5 VIA PERIMETRAL AV. CASUARINA COOP. BALERIO ESTACIO	38	10
FLORIDA NORTE MZ 114 SL 9	44	30
MAPASINGUE OESTE AV. 8VA CALLEJÓN 2DO	45	18
CDLA. GUAYAQUIL MZ. 9 VILLA 3	55	26
CDLA. MIRAFLORES LINDEROS #418 Y CALLE CUARTA	56	14
MIRAFLORES #115 Y CALLE SEGUNDA	46	27
CEIBOS NORTE	52	10
KENNEDY NORTE AV. ALEJANDRO COELLO Y JUAN ROLANDO	57	30
KENNEDY NORTE MZ 409 VILLA 2	56	24

CDLA. KENNEDY VIEJA 8VA 121 Y AV. SAN JORGE	48	14
KENNEDY NORTE CORNEJO MZ.91 S.14-15	57	28
TERCERA OESTE 103 Y AV. SAN JORGE	51	17
URDESA CENTRAL CALLE 5TA 206 B Y BALSAMOS	50	27

Tabla 4. 1 Datos para la Metaheurística

Fuente: Creado por el autor

Para la transformación de coordenadas geográficas a coordenadas rectangulares (UTM) se hizo uso de una aplicación sencilla pero muy efectiva para este trabajo como es la CALCULADORA UTM < > GEO versión 9.200712, la cual no entrega los datos resultantes en metros.

Los resultados obtenidos fueron los siguientes:

N°	Coordenadas Geográficas	X utm (m.)	Y utm (m.)	Zona	Hemisferio
1	-2.152634, -79.894920	5.165.970.327.624.610	1.130.021.852.208.680	30	sur
2	-2.130860, -79.891981	51.702.838.014.852.000	11.303.435.626.709.900	30	sur
3	-2.134877, -79.897133	51.694.113.615.533.300	11.297.697.585.740.900	30	sur
4	-2.098997, -79.905474	517.629.284.628.986	11.288.282.280.820.500	30	sur
5	-2.131269, -79.928840	51.695.895.665.821.400	11.262.301.210.284.100	30	sur
6	-2.137408, -79.896093	5.168.932.971.890.000	1.129.886.561.073.330	30	sur
7	-2.120402, -79.887381	5.172.410.199.697.300	11.308.538.544.233.400	30	sur
8	-2.118621, -79.913046	51.723.254.518.693.600	11.279.890.556.277.100	30	sur
9	-2.130871, -79.891810	51.702.844.966.686.400	11.303.626.499.160.100	30	sur
10	-2.146723, -79.893774	51.671.467.654.088	11.301.480.576.033.400	30	sur
11	-2.145376, -79.913984	51.670.793.634.374.500	11.278.921.743.530.900	30	sur
12	-2.133969, -79.933650	516.898.262.735.843	11.256.940.969.999.900	30	sur
13	-2.149639, -79.901129	5.166.455.649.501.440	1.129.328.053.245.370	30	sur
14	-2.133948, -79.933693	5.168.986.010.382.720	1.125.689.291.978.850	30	sur
15	-2.133937, -79.933640	51.689.890.369.609.100	112.569.520.373.586	30	sur
16	-2.129217, -79.923113	51.700.857.671.513.500	112.686.867.082.596	30	sur
17	-2.157413, -79.891511	51.650.894.319.600.400	11.304.036.666.483.700	30	sur
18	-2.138537, -79.923260	51.682.630.380.178.300	11.268.549.746.490.300	30	sur
19	-2.134012, -79.933629	5.168.974.586.468.650	11.256.964.531.465.400	30	sur
20	-2.195759, -79.901325	5.157.425.345.322.570	11.293.190.123.520.800	30	sur
21	-2.161482, -79.934666	5.163.598.231.954.210	11.255.885.703.723.200	30	sur
22	-2.891332, -78.984567	5.023.179.509.679.440	12.317.462.776.062.500	30	sur
23	-2.089419, -79.929393	5.177.749.037.237.940	11.261.559.152.574.300	30	sur

24	-2.186508, -79.892999	5.159.365.953.547.740	11.302.457.106.291.800	30	sur
25	-2.127339, -79.941284	51.701.484.232.460.400	11.248.401.893.337.100	30	sur
26	-2.128884, -79.937132	5.169.916.579.074.560	11.253.040.153.257.500	30	sur
27	-2.157955, -79.919793	51.645.265.179.871.400	11.272.474.550.978.700	30	sur
28	-2.145488, -79.945872	5.166.534.845.417.010	11.243.334.052.918.900	30	sur
29	-2.128269, -79.933659	5.170.094.598.065.800	11.256.914.318.972.200	30	sur
30	-2.250825, -79.892912	5.146.767.979.026.050	11.302.723.361.036.900	30	sur
31	-2.154365, -79.888851	5.165.729.728.279.200	11.306.996.647.912.600	30	sur
32	-2.163293, -79.920738	5.163.468.435.363.240	11.271.434.941.132.100	30	sur
33	-2.890131, -78.990836	50.234.225.199.309.500	12.310.465.594.734.700	30	sur
34	-2.383555, -80.355735	51.152.909.233.127.700	1.078.647.157.831.200	30	sur
35	-2.168887, -79.894214	5.162.798.379.245.140	11.301.052.349.023.100	30	sur
36	-2.160250, -79.897323	5.164.440.012.478.850	11.297.558.315.980.200	30	sur
37	-2.173567, -79.899573	5.161.796.785.157.860	11.295.084.562.165.100	30	sur
38	-2.161484, -79.896202	5.164.216.413.218.960	11.298.812.873.800.400	30	sur
39	-2.171715, -79.898688	51.621.733.990.065.900	1.129.606.709.987.950	30	sur
40	-2.172102, -79.909147	5.161.931.567.576.690	11.284.395.548.026.400	30	sur

Tabla 4. 2 Coordenadas Rectangulares

FASE CONSTRUCTIVA

En la primera fase de la metodología GRASP se genera una primera solución la cual sigue el siguiente proceso:

En el primero conjunto de solución que le llamaremos SL están todos los nodos que formaran parte de la solución inicial, de estos nodos se elige uno que ingresará a la solución, para esto se hace uso de la función glotona que hace que cada uno de los nodos candidatos a la función objetivo dentro de la lista SL vaya variando con cada iteración en la fase constructiva, luego se procede a obtener un intervalo de confianza:

$$Lmin = min;$$

$$Lmax = (1 - \alpha) * max + \alpha * min;$$

Donde *min* y *max* se refieren a los valores mínimo y máximo obtenido luego de la evaluación de todos los candidatos en la función glotona (Greddy), como es previsible estos valores van variando en cada una de las iteraciones de la fase constructiva.

Luego se obtiene la RL, esta lista está conformada por todos aquellos candidatos que luego de ser evaluados en la función Greddy, el valor obtenido cae dentro del rango de (L_{min} ; L_{max}) y que además la demanda no sobrepase la capacidad de los motorizados.

Por último, de la lista restringida (RL) se elige de manera aleatoria el nuevo integrante de la solución. Este proceso se repite hasta que la lista (SL) esté totalmente vacía.

Los principales datos utilizados en esta fase constructiva son:

- Las coordenadas de cada cliente y del Almacén, obtenidas a partir de la dirección proporcionada por la empresa y su transformación a coordenadas rectangulares.

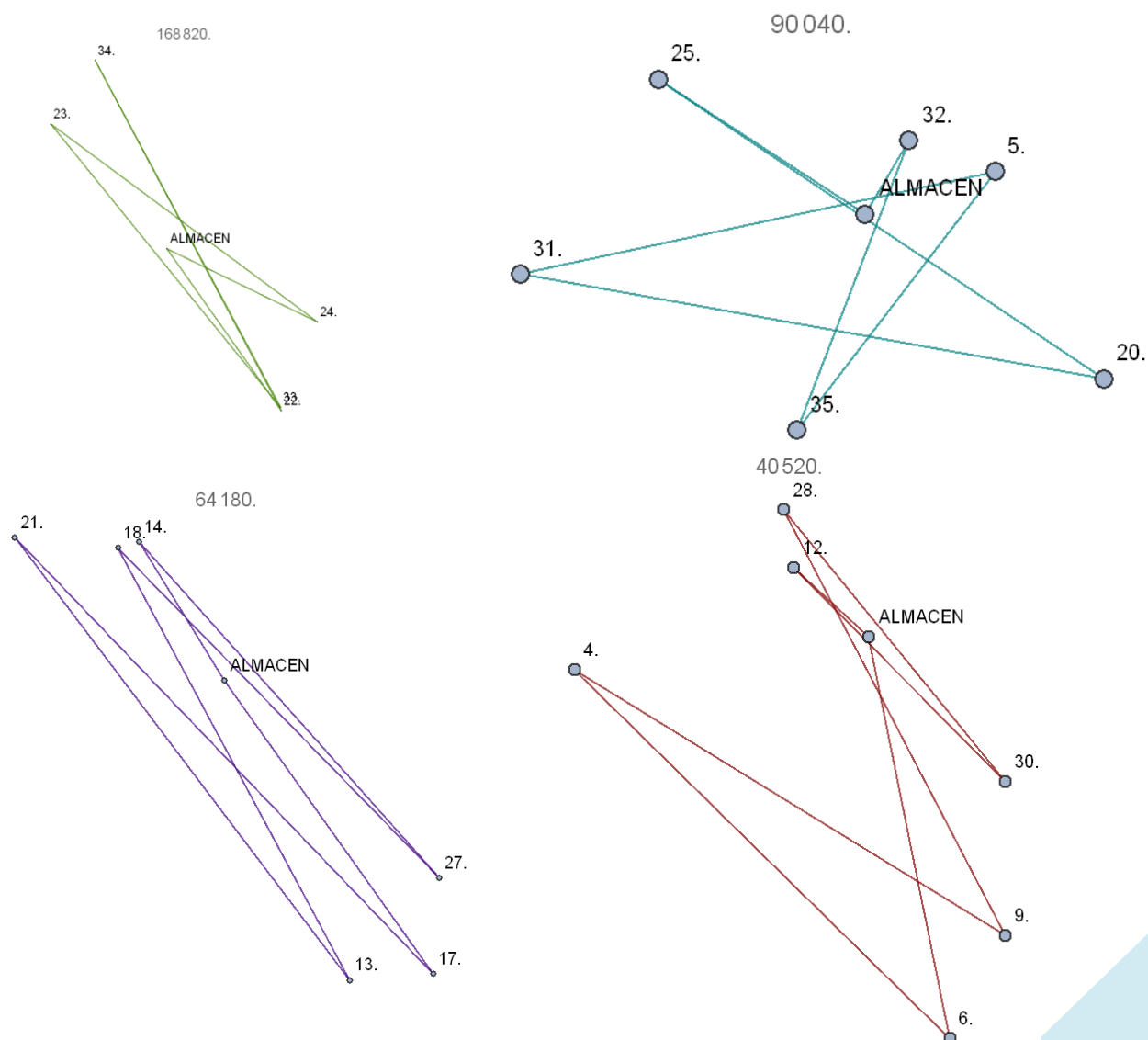
Las tablas correspondientes a coordenadas se adjuntaran en los Anexos respectivamente. Luego de realizar la programación del algoritmo GRASP en Mathematica 11, se obtuvieron los siguientes resultados mostrados en la tabla.

RUTAS	SECUENCIA GENERADA SIN MEJORA	DISTANCIAS
1	1-22-34-33-23-24-1	168.820
2	1-25-20-31-5-35-32-1	90.040
3	1-17-21-13-18-27-14-1	64.180
4	1-6-4-9-28-30-12-1	40.520
5	1-10-29-8-7-15-2-1	30.340
6	1-11-38-37-16-39-19-26-36-40-3-1	23.380
	TOTAL DISTANCIA EN MTS.	417.280

Tabla 4. 3 Resultados GRASP

Fuente: Creado por el autor

A continuación se muestran las rutas generadas por el Software.



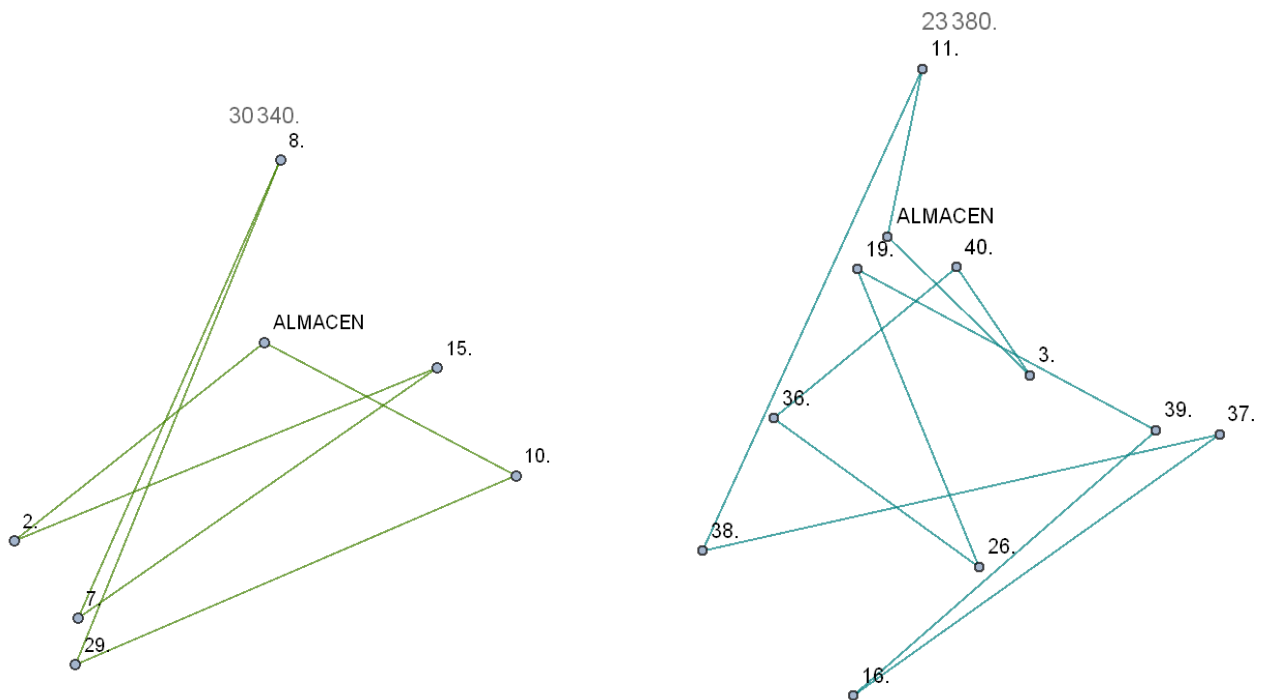


Gráfico 4. 2 Rutas resultantes GRASP

Fuente: Creado por el autor

En las figuras anteriores podemos identificar que en la fase constructiva nos genera una solución factible del problema, pero sin considerar los diferentes criterios de optimalidad, estos criterios son considerados en la siguiente fase que es la de mejora, en la cual trata que la búsqueda local encuentre una solución que mejore las soluciones encontradas en la fase constructiva.

Esto sucede en primer lugar porque los algoritmos basados en GRASP no dependen de la primera solución como si dependen otras metaheurísticas, sino de la cantidad de iteraciones que utilice el programa para correr, por lo tanto a mayores números de iteraciones, se tiene una alta probabilidad de alcanzar un mejor resultado.

FASE DE MEJORA

En esta fase, luego de completarse la construcción se trata de mejorar la solución obtenida con un método de búsqueda local, para este caso en específico se hizo uso del 2 opt, el cual nos permite intercambiar 2 nodos de la solución.

Este intercambio puede seguir hasta obtener la primera solución que sea mejor que la obtenida por la fase constructiva o la mejor de todas las soluciones obtenidas luego de realizar todos los intercambios permitidos por la heurística 2 opt, la diferencia entre estos dos criterios de parada radica en el tiempo y capacidad computacional necesarios para obtener los resultados, en el primer caso el criterio del primer mejor es mucho más rápido que el criterio del mejor de toda la exploración 2-OPT.

RUTAS	SECUENCIA GENERADA CON MEJORA	DISTANCIAS
1	1-13-17-20-33-22-24-1	41.600
2	1-27-9-6-35-29-2-36-1	22.680
3	1-8-14-18-21-34-23-25-1	40.760
4	1-15-5-32-11-12-28-1	14.600
5	1-39-37-10-30-26-3-40-1	7.520
6	1-19-31-4-38-7-16-1	20.543
	TOTAL DISTANCIA EN MTS.	147.703

Tabla 4. 4 Resultados Fase de mejora 2-OPT

Fuente: Creado por el autor

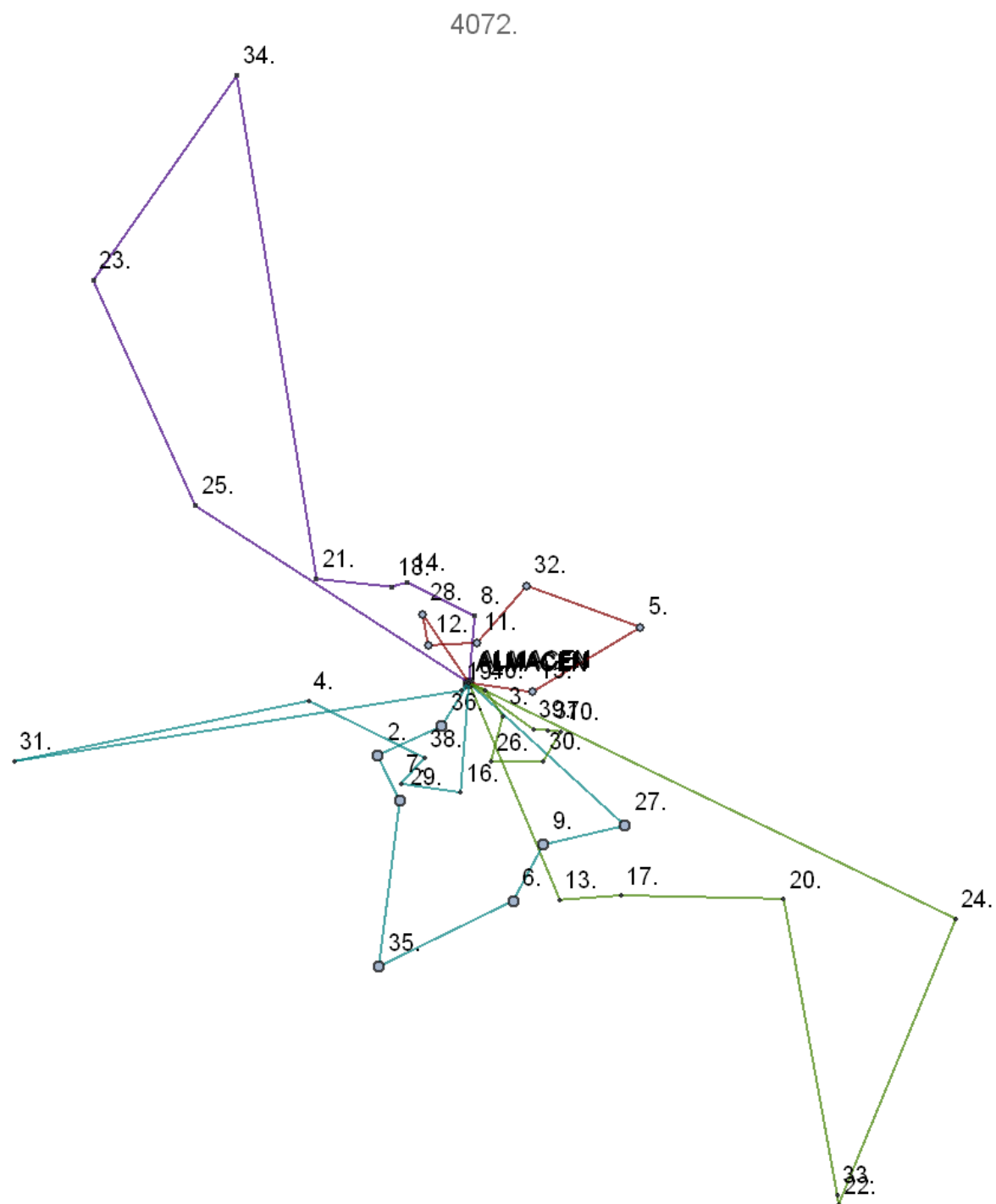


Gráfico 4. 3 Rutas fase de Mejora

Fuente: Creado por los autores

ANÁLISIS Y COMPARACIÓN DE RESULTADOS

	FASE CONSTRUCTIVA	FASE DE MEJORA	
RUTA	MTs	MTs	% AHORRO
1	168.820	41.600	75.4%
2	90.040	22.680	74.8%
3	64.180	40.760	36.4%
4	40.520	14.600	63.9%
5	30.340	7.520	75.2%
6	23.380	20.543	12.1%
	417.280	147.703	64.6%

Tabla 4. 5 Comparación entre F. Constructiva vs F. Mejora

Fuente: Creado por el autor

Al analizar y comparar los resultados de ambas fases podemos notar que la fase constructiva es mejorada en gran medida por el algoritmo 2-OPT, en cada una de las rutas generadas como se muestra en la tabla 4.6, además en el recorrido total en metros tiene una mejora porcentual de 64.6%.

Finalmente para la comparación entre la operación que realiza actualmente la empresa ABC y lo que se obtuvo en el algoritmo, primero se analizó los criterios de las personas encargadas de realizar las rutas de cobranza anteriormente, esto se le llevaba aproximadamente medio día trabajo y por lo general las rutas eran realizadas para el día siguiente.

DISTANCIAS RECORRIDAS	
CON RUTEO ACTUAL	CON RUTEO OPTIMIZADO
417280	147703

Tabla 4. 6 Distancias Recorridas

Fuente: Creado por el autor

CONCLUSIONES

Este proyecto se lo realizó con el propósito de establecer un sistema de ruteo para optimizar las visitas a los clientes, los cuales se les realizará el cobro de sus facturas, ya que actualmente la empresa asigna las rutas de forma empírica.

Con la metaheurística basada en la metodología GRASP, desarrollada en este proyecto, se cumplió con el objetivo de visitar a todos los clientes seleccionados, minimizando las distancias recorridas.

Los tiempos de generación de rutas se disminuyeron notoriamente a un aproximado de 15 minutos y las distancias totales recorridas en un 64,6%, frente a lo que tardaba la empresa en generar las rutas que era medio día laborable del día anterior y en ocasiones al incumplimiento de llegar a visitar a todos los clientes.

En conclusión, el algoritmo 2-OPT fue de gran ayuda como método de mejora local, se logró observar que la fase constructiva solo tenía como fin encontrar una solución inicial que fuese factible, luego el algoritmo 2opt se encarga de mejorar drásticamente la solución inicial.

Por último, al disminuir las distancias recorridas hacen que las operaciones de los motorizados sean más ágiles, ya que para ellos es un problema el recoger dinero por ser una tarea de alto riesgo, por la inseguridad que se vive en los actuales momentos, como también reducir la emisión de CO₂ emitida hacia la atmósfera.

RECOMENDACIONES

Para proyectos futuros relacionados con la metaheurística GRASP, para resolver el VRPTW, se recomienda utilizar otros métodos de post-optimización como lo son: Path Relinking, Colonias de hormigas, o generalizar este método y desarrollar un Recocido Simulado con las mejores soluciones obtenidas en el procedimiento GRASP.

Se recomienda utilizar como criterio de parada el recorrer todos los vecinos y de ellos obtener el mejor en el procedimiento 2opt, porque aunque computacionalmente hablando es mucho más pesado y toma más tiempo, realmente brinda mejores soluciones que el criterio de parada del primer mejor.

En el presente trabajo, se ha considerado un modelo de rutas con flota de capacidad homogénea pero se puede extender a un problema de capacidad heterogénea, con esto llevaría a tener un parámetro más en el problema planteado, que sería la capacidad de cada camión o la demanda de cada cliente.

Finalmente es importante mencionar que cualquier proceso de enrutamiento debe ser monitoreado por un dispositivo móvil, por temas de seguridad la cual ayuda a brindar coordenadas exactas de dónde y cómo se están movilizand los motorizados con el dinero.

BIBLIOGRAFÍA

- [1] P.K. Nguyen, T.G. Craynic, M. Toulouse. (2014), Toronto, Canada. A hybrid generational genetic algorithm for the periodic vehicle routing problem with time Windows
- [2] Zhiye Li, Songshan Guo, Fan Wang, Andrew Lim. (2014), Hong Kong-China. Improved GRASP with Tabu search for Vehicle Routing with Both Time Window and Limited Number of Vehicles.
- [3] Elizabeth Mancera-Galván, Beatriz A. Garro-Licón, Katya Rodríguez-Vázquez IIMAS-UNAM, Ciudad Universitaria, D.F., México (2015). Optimización mediante algoritmo de hormigas aplicado a la recolección de residuos sólidos.
- [4] Fernando Sandoya 2014, Metaheurística y Redes Neuronales.
- [5] GARCÍA-NAJERA, A.; BULLINARIA, J. A. (2011), «An improved multi objective evolutionary algorithm for the vehicle routing problem with time windows». Computers & Operations Research, vol. 38 (1), pp. 287-300.
- [6] MATOS, A. C.; OLIVEIRA, R. C. (2004), Aplicación de la metodología GRASP al problema de Rutificación de Vehículos, Application of GRASP methodology to Vehicle Routing Problem.

ANEXOS

Guayaquil, 23 de Mayo del 2016

Cuestionario-Entrevista

- ✓ Departamento de Facturación y Cobranzas, en cual realizare el proyecto de planificación de rutas en la empresa ABC.
- ✓ Persona entrevistada es:
DENIS CRSITINA CAJAS ARENAS
- ✓ Para el desarrollo de la entrevista, utilice la estructura en forma de embudo, la cual empieza con preguntas abiertas y finalizo con preguntas cerradas.

Preguntas:

1. Comentarios acerca de la empresa ABC.
2. ¿Cuánto tiempo tiene a cargo de la gerencia de la empresa?

Aproximadamente 2 años.

3. ¿Cómo realiza el Servicio de cobranza a sus clientes?

Se genera la factura, y de manera empírica se asigna la cantidad de clientes para cada recolector.

4. ¿Cuáles son los principales problemas que usted ha determinado en la recolección de dinero?

El incumplimiento de visitar a todos los clientes asignados, y por ende pérdida de flujo de efectivo o malestar por parte del cliente.

5. ¿Cuál es el tamaño de flota que posee para dicha operación?

3 motorizados.

6. Considera que la atención brindada por ustedes al cliente es la adecuada?

Sí, pero podría ser mejor.

7. ¿Consideraría usted necesario realizar algún tipo de estudio acerca de una buena Planificación de rutas para tratar de optimizarlo?

Sí.

8. ¿En qué sucursal estaría interesado que realice este proyecto?

En Guayaquil, Matriz ubicado en la cdla. La Garzota.

9. ¿Qué expectativas tiene usted respecto a la investigación que se implementará en la empresa?

Optimización de distancias recorridas, visitar a todos los clientes y obtener mejores resultados de ingreso de efectivo.

10. ¿Cree usted necesario realizar algún tipo de investigación con el objetivo de resolver el problema encontrado en el la Planificación de Rutas? (sí o No)

Sí

11. ¿Está de acuerdo con nuestra intervención en el proyecto de mejora en las Rutas? (si – no)

Sí

Código de programación del algoritmo con metodología GRASP

(Mathematica-11):

```
datos = Import[
    "C:\\Users\\campoverde\\Desktop\\DATOS-JONATHAN.xlsx", {"Data",
4}];
datos = Delete[datos, 1];
coord = Table[{datos[[i, 2]], datos[[i, 3]]}, {i, Length[datos]}];
d = Table[
    N[ManhattanDistance[datos[[i, 2 ;; 3]], datos[[j, 2 ;; 3]]],
    4], {i, Length[datos]}, {j, Length[datos]}];
base = {datos[[1, 2]], datos[[1, 3]]};
n = 40;
tour = {1};
noselec = Range[2, n];
longitud = 0;
k = 1;(*k es el ultimo vertice explorado*)
long[x_] := Sum[d[[x[[i]], x[[i + 1]]]], {i, 1, n}];
While[Length[tour] <= n - 1, distr = Table[d[[k, i]], {i, noselec}];
    m = Max[distr];
    longitud += m;
    k = noselec[[Position[distr, m][[1, 1]]]];
    AppendTo[tour, k];
    noselec = DeleteCases[noselec, k];
    AppendTo[tour, 1];
```

```

longitud += d[[k, 1]];

w = {}; ww = {};

ff = Partition[tour, 6];

aa = AppendTo[ff[[1 ;; 5]],

  Join[ff[[6]],

    Flatten[tour[[(Length[ff]*Length[ff[[1]])] + 1 ;;

      Length[tour]]]]];

For[i = 1, i <= Length[aa], i++,

  AppendTo[w, DeleteDuplicates[PrependTo[aa[[i]], 1]]];

For[i = 1, i <= Length[aa], i++, AppendTo[ww, AppendTo[w[[i]], 1]]];

distancia2[lista_] :=

  N[Sum[d[[lista[[i]], lista[[i + 1]]], {i, 1, Length[lista] - 1}]];

long = Table[distancia2[ww[[m]]], {m, Length[ww]}];

tourorg = ww;

mejororg = long;

numrutas = Length[ww] + 1;

If[Length[ww] < numrutas, numrutas = Length[ww]; mejorrutas = ww;

Do[Print["Ruta ", j, " : ", mejorrutas[[j]], "  ",

  "Distancia recorrida : ", distancia2[mejorrutas[[j]]], {j,

  Length[mejorrutas]}] Print[

"DISTANCIA TOTAL RECORRIDA EN TODAS LAS RUTAS: ",

Sum[d[[ww[[i, j]], ww[[i, j + 1]]], {i, Length[ww]}, {j,

  Length[ww[[i]]] - 1}] " CON EL NUMERO DE RUTAS DE: ",

numrutas];

```

```

a = {};
a = Table[
  Graph[Table[
    mejorrutas[[j]][[i]] <-> mejorrutas[[j]][[i + 1]], {i,
      Length[mejorrutas[[j]] - 1}},
    VertexCoordinates ->
      datos[[mejorrutas[[j]][[1 ;; -2]]][[All, 2 ;; 3]],
    VertexLabels ->
      Append[Table[
        mejorrutas[[j]][[i]] -> datos[[mejorrutas[[j]][[i, 1]], {i,
          2, Length[mejorrutas[[j]] - 1}}, 1 -> "ALMACEN"],
        VertexSize -> Medium, EdgeStyle -> Hue[j/4, 1, 1/2],
        PlotLabel -> distancia2[mejorrutas[[j]]], {j,
          Length[mejorrutas]}];
    Print[Column[{"Rutas", TableForm[a]}, Center, Frame -> All]];
    Print[Show[a]]
    Print["ETAPA DE MEJORA"]
    cap = 4500;
    n = Length[datos];
    rutas = Table[{1, i, 1}, {i, 2, n}];
    s[a_, b_] := d[[1, a[[2]]]] + d[[1, b[[2]]]] - d[[a[[2]], b[[2]]]];
    carga[a_] := Sum[datos[[a[[i]], 4]], {i, 2, Length[a] - 1}];
    vel = 40;
    tiempomovimiento = d/vel;

```

```

Column[{"Tabla de ahorros (savings)",
TableForm[
  ahorros =
    Table[s[rutas[[i]], rutas[[j]], {i, n - 1}, {j, 1, i - 1}],
    TableHeadings -> {datos[[2 ;; n, 1]], datos[[2 ;; n, 1]]}],
Center, Frame -> All];
opcion = True;
matrixtiempo =
  Table[{datos[[i, 5]], datos[[i, 7]], datos[[i, 6]]}, {i, 2, n}];
While[Max[ahorros] > 0, (*Se escoge el mayor ahorro*) m = Max[ahorros];
  (*Busca que clientes pertenecen al ahorro actual*) {p, q} =
    Position[ahorros, m][[1]];
  (*Busca en que ruta se encuentra los clientes actuales*)
  pp = Position[rutas, p + 1][[1]];
  qq = Position[rutas, q + 1][[1]];
  (*Si los clientes estan ya en la misma ruta(verdadero),
  sino estan empiezan a buscar si se puede o no se puede agregar*)
  If[pp[[1]] == qq[[1]], ahorros[[p, q]] = 0,
  Which[(*sobrepasa capacidad*)
    carga[Flatten[
      Insert[rutas[[pp[[1]]], Reverse[rutas[[qq[[1]], 2 ;; -2]],
        2]]] >= cap,
    ahorros[[p, q]] =
      0, (*los clientes del ahorro se encuentran en la 2da posicion*)

```

```

pp[[2]] == 2 && qq[[2]] == 2 &&
carga[Flatten[
  Insert[rutas[[pp[[1]]]], Reverse[rutas[[qq[[1]], 2 ;; -2]]],
  2]] < cap,
posibleruta =
Flatten[Insert[rutas[[pp[[1]]]],
  Reverse[rutas[[qq[[1]], 2 ;; -2]]], 2];
boleana = 1;
a = 0;
Do[If[
  Max[a, matrixtiempo[[posibleruta[[i + 1]] - 1, 1]] +
    matrixtiempo[[posibleruta[[i + 1]] - 1, 3]] +
    tiempomovimiento[[pp, qq]] >
    matrixtiempo[[posibleruta[[i + 1]] - 1, 2]], boleana = 0,
  a = Max[a, matrixtiempo[[posibleruta[[i + 1]] - 1, 1]] +
    matrixtiempo[[posibleruta[[i + 1]] - 1, 3]] +
    tiempomovimiento[[pp, qq]]], {i, Length[posibleruta] - 2};
boleana2 = 0;
a = 0;
If[boleana == 0, boleana2 = 1;
Do[If[
  Max[a, matrixtiempo[[posibleruta[[i]] - 1, 1]] +
    matrixtiempo[[posibleruta[[i]] - 1, 3]] +
    tiempomovimiento[[pp, qq]] >

```

```

matrixtiempo[[posibleruta[[i]] - 1, 2]], booleana2 = 0,
a = Max[a, matrixtiempo[[posibleruta[[i]] - 1, 1]] +
matrixtiempo[[posibleruta[[i]] - 1, 3]] +
tiempomovimiento[[pp, qq]], {i, Length[posibleruta] - 1,
2, -1}];];
If[booleana == 1,
AppendTo[rutas,
Flatten[Insert[rutas[[pp[[1]]]],
Reverse[rutas[[qq[[1]], 2 ;; -2]], 2]]];
rutas = Delete[rutas, {{pp[[1]]}, {qq[[1]]}}];];
If[booleana2 == 1,
AppendTo[rutas,
Reverse[Flatten[
Insert[rutas[[pp[[1]]]], Reverse[rutas[[qq[[1]], 2 ;; -2]],
2]]];];
rutas = Delete[rutas, {{pp[[1]]}, {qq[[1]]}}];];
ahorros[[p, q]] =
0,(*el cliente 1 se encuentra en la 2da posicion y el cliente 2 \
se encuentra en la penultima*)
pp[[2]] == 2 && qq[[2]] == rutas[[qq[[1]], -2]] &&
carga[Flatten[
Insert[rutas[[pp[[1]]]], rutas[[qq[[1]], 2 ;; -2]], 2]] <=
cap, posibleruta =
Flatten[Insert[rutas[[pp[[1]]]], rutas[[qq[[1]], 2 ;; -2]], 2]]];

```

```

booleana = 1;

a = 0;

Do[If[

  Max[a, matrixtiempo[[posibleruta[[i + 1]] - 1, 1]] +

    matrixtiempo[[posibleruta[[i + 1]] - 1, 3]] +

    tiempomovimiento[[pp, qq]] >

    matrixtiempo[[posibleruta[[i + 1]] - 1, 2]], booleana = 0,

  a = Max[a, matrixtiempo[[posibleruta[[i + 1]] - 1, 1]] +

    matrixtiempo[[posibleruta[[i + 1]] - 1, 3]] +

    tiempomovimiento[[pp, qq]], {i, Length[posibleruta] - 2}];

booleana2 = 0;

a = 0;

If[booleana == 0, booleana2 = 1;

Do[If[

  Max[a, matrixtiempo[[posibleruta[[i]] - 1, 1]] +

    matrixtiempo[[posibleruta[[i]] - 1, 3]] +

    tiempomovimiento[[pp, qq]] >

    matrixtiempo[[posibleruta[[i]] - 1, 2]], booleana2 = 0,

  a = Max[a, matrixtiempo[[posibleruta[[i]] - 1, 1]] +

    matrixtiempo[[posibleruta[[i]] - 1, 3]], {i,

Length[posibleruta] - 1, 2, -1}];];

If[booleana == 1,

AppendTo[rutas,

Flatten[Insert[rutas[[pp[[1]]]], rutas[[qq[[1]]], 2 ;; -2]],

```

```

2]] + tiempomovimiento[[pp, qq]];
rutas = Delete[rutas, {{pp[[1]]}, {qq[[1]]}}];
If[booleana2 == 1,
AppendTo[rutas,
Reverse[Flatten[
Insert[rutas[[pp[[1]]], rutas[[qq[[1]], 2 ;; -2]], 2]]];
rutas = Delete[rutas, {{pp[[1]]}, {qq[[1]]}}];
ahorros[[p, q]] =
0,(*el cliente 2 se encuentra en la 2da posicion y el cliente 1 \
se encuentra en la penultima*)(**)
pp[[2]] == rutas[[pp[[1]], -2]] && qq[[2]] == 2 &&
carga[
Flatten[Insert[rutas[[pp[[1]]],
rutas[[qq[[1]], 2 ;; -2]], -2]] <= cap,
posibleruta =
Flatten[Insert[rutas[[pp[[1]]], rutas[[qq[[1]], 2 ;; -2]], -2]];
booleana = 1;
a = 0;
Do[If[
Max[a, matrixtiempo[[posibleruta[[i + 1]] - 1, 1]]] +
matrixtiempo[[posibleruta[[i + 1]] - 1, 3]] +
tiempomovimiento[[pp, qq]] >
matrixtiempo[[posibleruta[[i + 1]] - 1, 2]], booleana = 0,
a = Max[a, matrixtiempo[[posibleruta[[i + 1]] - 1, 1]]] +

```

```

matrixtiempo[[posibleruta[[i + 1]] - 1, 3]] +
tiempomovimiento[[pp, qq]], {i, Length[posibleruta] - 2}];

boleana2 = 0;

a = 0;

If[boleana == 0, boleana2 = 1;

Do[If[
Max[a, matrixtiempo[[posibleruta[[i]] - 1, 1]]] +
matrixtiempo[[posibleruta[[i]] - 1, 3]] +
tiempomovimiento[[pp, qq]] >
matrixtiempo[[posibleruta[[i]] - 1, 2]], boleana2 = 0,
a = Max[a, matrixtiempo[[posibleruta[[i]] - 1, 1]]] +
matrixtiempo[[posibleruta[[i]] - 1, 3]], {i,
Length[posibleruta] - 1, 2, -1}];];

If[boleana == 1,
AppendTo[rutas,
Flatten[Insert[rutas[[pp[[1]]],
rutas[[qq[[1]], 2 ;; -2]], -2]] + tiempomovimiento[[pp, qq]]];
rutas = Delete[rutas, {{pp[[1]]}, {qq[[1]]}}];];

If[boleana2 == 1,
AppendTo[rutas,
Reverse[Flatten[
Insert[rutas[[pp[[1]]], rutas[[qq[[1]], 2 ;; -2]], -2]]];];
rutas = Delete[rutas, {{pp[[1]]}, {qq[[1]]}}];] ahorros[[p,
q]] = 0, (*Los clientes del ahorro se encuentran en la \

```

penultima posicion*)

```
pp[[2]] == rutas[[pp[[1]], -2]] &&
```

```
qq[[2]] == rutas[[qq[[1]], -2]] &&
```

```
carga[Flatten[
```

```
  Insert[rutas[[pp[[1]]],
```

```
    Reverse[rutas[[qq[[1]], 2 ;; -2]], -2]] <= cap,
```

```
posibleruta =
```

```
  Flatten[Insert[rutas[[pp[[1]]],
```

```
    Reverse[rutas[[qq[[1]], 2 ;; -2]], -2]];
```

```
booleana = 1;
```

```
a = 0;
```

```
Do[If[
```

```
  Max[a, matrixtiempo[[posibleruta[[i + 1]] - 1, 1]]] +
```

```
  matrixtiempo[[posibleruta[[i + 1]] - 1, 3]] +
```

```
  tiempomovimiento[[pp, qq]] >
```

```
  matrixtiempo[[posibleruta[[i + 1]] - 1, 2]], booleana = 0,
```

```
  a = Max[a, matrixtiempo[[posibleruta[[i + 1]] - 1, 1]]] +
```

```
  matrixtiempo[[posibleruta[[i + 1]] - 1, 3]] +
```

```
  tiempomovimiento[[pp, qq]], {i, Length[posibleruta] - 2}];
```

```
booleana2 = 0;
```

```
a = 0;
```

```
If[booleana == 0, booleana2 = 1;
```

```
  Do[If[
```

```
    Max[a, matrixtiempo[[posibleruta[[i]] - 1, 1]]] +
```

```

matrixtiempo[[posibleruta[[i]] - 1, 3]] +
tiempomovimiento[[pp, qq]] >
matrixtiempo[[posibleruta[[i]] - 1, 2]], boleana2 = 0,
a = Max[a, matrixtiempo[[posibleruta[[i]] - 1, 1]]] +
matrixtiempo[[posibleruta[[i]] - 1, 3]] +
tiempomovimiento[[pp, qq]], {i, Length[posibleruta] - 1,
2, -1}];];
If[boleana == 1,
AppendTo[rutas,
Flatten[Insert[rutas[[pp[[1]]]],
Reverse[rutas[[qq[[1]], 2 ;; -2]], -2]]];
rutas = Delete[rutas, {{pp[[1]]}, {qq[[1]]}}];];
If[boleana2 == 1,
AppendTo[rutas,
Reverse[Flatten[
Insert[rutas[[pp[[1]]]],
Reverse[rutas[[qq[[1]], 2 ;; -2]], -2]]];
rutas = Delete[rutas, {{pp[[1]]}, {qq[[1]]}}];];
ahorros[[p, q]] =
0,(*Uno de los dos no se encuentra en la 2da o penultima \
posicion*)(pp[[2]] != 2 &&
pp[[2]] != rutas[[pp[[1]], -2]] || (qq[[2]] != 2 &&
qq[[2]] != rutas[[qq[[1]], -2]]), ahorros[[p, q]] = 0)];
distancia2[lista_] :=

```

```

N[Sum[d[[lista[[i]], lista[[i + 1]]], {i, 1, Length[lista] - 1}]];
long = Table[distancia2[rutas[[m]]], {m, Length[rutas]};
tourorg = rutas;
mejororg = long;
numrutas = Length[rutas] + 1;
If[Length[rutas] < numrutas, numrutas = Length[rutas];
mejorrutas = rutas;
Do[Print["Ruta ", j, " : ", mejorrutas[[j]], "  ",
"Distancia Recorrida : ", distancia2[rutas[[j]]],
" Demanda recogida : ", carga[mejorrutas[[j]]], {j,
Length[mejorrutas]}] Print[
"DISTANCIA TOTAL RECORRIDA EN TODAS LAS RUTAS: ",
Sum[d[[rutas[[i, j]], rutas[[i, j + 1]]], {i, Length[rutas]}, {j,
Length[rutas[[i]]] - 1}] " CON EL NUMERO DE RUTAS DE: ",
numrutas];
a = {};
a = Table[
Graph[Table[
mejorrutas[[j]][[i]] <-> mejorrutas[[j]][[i + 1]], {i,
Length[mejorrutas[[j]]] - 1},
VertexCoordinates ->
datos[[mejorrutas[[j]][[1 ;; -2]]][[All, 2 ;; 3]],
VertexLabels ->
Append[Table[

```

```
mejorrutas[[j]][[i]] -> datos[[mejorrutas[[j]]][[i, 1]], {i,  
2, Length[mejorrutas[[j]] - 1}}, 1 -> "ALMACEN",  
VertexSize -> Medium, EdgeStyle -> Hue[j/4, 1, 1/2],  
PlotLabel -> carga[mejorrutas[[j]]], {j, Length[mejorrutas]};  
Print[Column[{"Rutas", TableForm[a]}, Center, Frame -> All];  
Print[Show[a]]]
```