

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Máster Universitario en Inteligencia Artificial, Reconocimiento
de Formas e Imagen Digital



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

**Diseño y aplicación de técnicas
inteligentes en problemas de logística**

Autor:

Ing. Juan José Ponce Torres

Tutor:

Phd. Antonio Garrido Tejero

Curso 2018 -2019

A mis abuelos Guillermo y Rómulo.

Agradecimientos

Quisiera agradecer:

- A Dios, por ser el motor que me impulsa a diario y me ha permitido obtener un logro más.
- A la Universitat Politècnica de València, por abrirme sus puertas y brindarme la oportunidad estudiar esta maestría.
- A mis maestros, por su calidad humana y por compartir íntegramente sus conocimientos.
- A mi profesor guía por su labor en este trabajo investigativo.
- A mis padres, por brindarme su apoyo incondicional y motivarme a siempre seguir adelante.
- A mis hermanos, por impulsarme a ser mejor.
- A Sergio, por su amistad y su apoyo incondicional.

Resumen

En este trabajo de investigación se emplea algoritmos genéticos aplicados a los problemas de logística. El algoritmo es implementado para optimizar los resultados obtenidos en la distribución de productos para un grupo de tiendas que pertenecen una cadena de supermercados. Este documento presenta el diseño utilizado en la implementación del algoritmo para que se pueda cumplir con las restricciones asociadas al problema. Además, un meta algoritmo fue creado para obtener la adecuada selección de la configuración inicial que requiere esta heurística. Se realizaron diversas evaluaciones probando distintas configuraciones, los resultados generados están descritos en el documento. Adicional, se creó una plataforma web para que se ingresen todos los datos que necesita el algoritmo diseñado para la generación de las rutas. Esta aplicación permitirá a los usuarios entender, de una forma más sencilla, los resultados generados para el reparto de productos.

Abstract

This research work uses genetic algorithms applied to logistics problems. The algorithm is implemented to optimize the results obtained in the distribution of products for a group of stores that belong to a supermarket chain. This document presents the algorithm design used in the implementation so that it can comply with the restrictions associated with the problem. In addition, a meta algorithm was created to obtain the appropriate selection of the initial configuration required by this heuristic. Various evaluations were obtained by testing different configurations, the results generated are determined in the document. In addition, a web platform was created to insert all the data required by the designed algorithm for the generation of routes. This application allowed users to understand in a simpler way the results generated for the distribution of products.

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Antecedentes y problemática	2
1.2.1. Entidades asociadas al problema	4
1.3. Objetivos	8
1.4. Estructura del Documento	9
2. Fundamentos teóricos	10
2.1. Problema de ruteo de vehículo (VRP)	10
2.1.1. Componentes de problema de ruteo de vehículos	10
2.1.2. Funciones objetivo	11
2.1.3. Variantes del problema	11
2.1.4. Restricciones adicionales	12
2.2. Optimización para problemas multiobjetivo	13
2.2.1. Métodos de resolución de problemas multiobjetivo	13
2.3. Algoritmos Genéticos	14
2.3.1. Codificación	15
2.3.2. Población	15
2.3.3. Función de evaluación	16
2.3.4. Restricciones	16
2.3.5. Selección	17
2.3.6. Cruce	18
2.3.7. Mutación	18
2.3.8. Reducción	19
2.3.9. Finalización	19
2.3.10. Experimentación	19
2.3.11. Multimodal	19
2.3.12. Selección de parámetros	21

3. Construcción del algoritmo genético	23
3.1. Diseño del algoritmo en base al problema	23
3.1.1. Configuración inicial	23
3.1.2. Gen	25
3.1.3. Cromosoma	27
3.1.4. Individuo	28
3.1.5. Población	28
3.2. Creación de una solución	30
3.3. Evaluación del individuo	31
3.3.1. Evaluación de la distancia total	31
3.3.2. Evaluación del número de vehículos utilizados	32
3.3.3. Evaluación del coste total de combustible	32
3.3.4. Evaluación del tiempo de reparto	33
3.3.5. Evaluación del coste del personal involucrado	34
3.4. Selección de los individuos padres	35
3.5. Operador de cruce	36
3.6. Operador de mutación	38
3.7. Reparar el gen	39
3.8. Reducción de población	40
3.9. Condición de parada	41
4. Aplicación web para la gestión de rutas	42
4.1. Bases de datos	42
4.1.1. Bases de datos adicionales	44
4.1.2. Fuentes de datos adicionales	45
4.2. Tecnologías utilizadas	47
4.3. Funcionamiento de la aplicación web	47
4.3.1. Pantalla de inicio	48
4.3.2. Tiendas	48
4.3.3. Productos	49
4.3.4. Vehículos	49
4.3.5. Conductores	50
4.3.6. Tipo de Vehículos	51
4.3.7. Permisos de conducir	52
4.3.8. Tipo de combustible	52
4.3.9. Ciudades	53
4.3.10. Provincias	53
4.3.11. Familia de productos	54
4.3.12. Eventos Festivos	55
4.3.13. Configuración	56

4.4.	Uso del algoritmo en la aplicación web	56
4.4.1.	Configuración de la búsqueda de rutas	56
4.4.2.	Resultados de la generación de las rutas	57
4.5.	Selección de parámetros en la aplicación web	58
5.	Evaluación y análisis de los resultados	60
5.1.	Evaluación algoritmo genético	60
5.1.1.	Evaluaciones realizadas	60
5.1.2.	Resultados obtenidos	61
5.2.	Evaluación meta algoritmo	62
5.2.1.	Diseño del meta algoritmo	62
5.2.2.	Evaluaciones realizadas sobre meta algoritmo	69
5.2.3.	Resultados obtenidos	70
5.3.	Evaluación algoritmo genético usando la mejor configuración	71
5.3.1.	Resultados obtenidos	71
5.4.	Análisis de los resultados	72
6.	Conclusiones y Recomendaciones	74
6.1.	Conclusiones	74
6.2.	Recomendaciones y trabajo futuro	75

Bibliografía

Índice de figuras

1.1. Representación del problema que se pretende resolver	4
2.1. Representación del cruce entre dos individuos padres	18
2.2. Representación de mutación de un individuo	19
3.1. Ejemplo de gen	26
3.2. Ejemplo de cromosoma	27
3.3. Ejemplo de individuo	28
3.4. Ejemplo de población	29
3.5. Ejemplo del proceso de selección de los individuos	35
3.6. Ejemplo de cruce	37
3.7. Ejemplo de mutación	39
3.8. Ejemplo de eliminar individuos	40
4.1. Modelo de la base de datos	46
4.2. Pantalla de inicio de la aplicación web	48
4.3. Pantalla de tiendas en aplicación web	49
4.4. Pantalla de productos en aplicación web	49
4.5. Pantalla de vehículos en aplicación web	50
4.6. Pantalla de conductores en aplicación web	51
4.7. Pantalla de tipos de vehículos en aplicación web	51
4.8. Pantalla de permisos de conducir en aplicación web	52
4.9. Pantalla de tipos de combustible en aplicación web	53
4.10. Pantalla de ciudades en aplicación web	53
4.11. Pantalla de provincias en aplicación web	54
4.12. Pantalla de familias de productos en aplicación web	55
4.13. Pantalla de días festivos en aplicación web	55
4.14. Pantalla de configuración general en aplicación web	56
4.15. Pantalla de configuración para generar rutas en aplicación web	57
4.16. Pantalla de rutas generadas en aplicación web	58
4.17. Pantalla de rutas generadas usando mapa en aplicación web	58

4.18. Pantalla para ingresar configuración inicial de meta algoritmo en aplicación web	59
5.1. Ejemplo de individuo usado por el meta algoritmo	64
5.2. Ejemplo de población usado por el meta algoritmo	65
5.3. Ejemplo del operador de selección en el meta algoritmo	66
5.4. Ejemplo del operador de cruce en el meta algoritmo	67
5.5. Ejemplo del operador de mutación en el meta algoritmo	67
5.6. Ejemplo de eliminación de individuos en el meta algoritmo	68

Índice de tablas

5.1. Tabla que muestra la configuración inicial en cada una de las evaluaciones realizadas con el algoritmo genético en la generación de rutas	61
5.2. Mejores valores de aptitud obtenidos en las evaluaciones del algoritmo genético para la generación de rutas	62
5.3. Parámetros usados en la primera evaluación del meta algoritmo	69
5.4. Parámetros usados en la segunda evaluación del meta algoritmo	69
5.5. Parámetros usados en la tercera evaluación del meta algoritmo	70
5.6. Parámetros usados en la cuarta evaluación del meta algoritmo	70
5.7. Mejores resultados obtenidos durante las evaluaciones realizadas con el meta algoritmo para la obtención de la configuración inicial	70
5.8. Resultados obtenidos usando la mejor configuración inicial encontrada por el meta algoritmo	72

Capítulo 1

Introducción

Las empresas cada vez se enfrentan a una mayor competitividad en los mercados globales y buscan maneras de innovarse para poder mantener una buena posición respecto a sus competidores. El manejo de la gestión de sus operaciones constantemente debe ser mejorado porque permite obtener ventajas en relación con sus competidores. Los sistemas operacionales logísticos y de cadena de suministro constantemente se modernizan para poder adaptarse a los cambios sociales y tecnológicos. Estas actualizaciones requieren que los investigadores y los profesionales capacitados los reconsideren e incorporen nuevas funcionalidades. A su vez, plantea nuevas demandas de personal cualificado para operar estos sistemas [1].

En los últimos años el uso de técnicas de inteligencia artificial ha crecido, porque está desempeñando un papel fundamental sobre diversas áreas entre las que se incluye la logística y gestión de la cadena de operaciones. En muchos casos se emplea en el procesamiento de datos no estructurados, por ejemplo, el procesamiento del lenguaje natural. También, existen técnicas de aprendizaje automático que usan datos estructurados para mejorar la toma de decisiones. Las técnicas mencionadas anteriormente no requieren definir y codificar todas las reglas asociadas al problema ya que se realizan en un análisis predicativo. Algunos ejemplos de estas técnicas son: las redes neuronales y las heurísticas [2].

1.1. Motivación

Los algoritmos genéticos aplicados a problemas de logística es un tema que se ha estudiado durante muchos años y ha tenido éxito en la implementación dentro de muchas empresas. Los investigadores constantemente han propuesto algoritmos más avanzados que permiten predecir información más precisa y dar mejores soluciones

a las versiones anteriores. Estos algoritmos deben ser utilizados en herramientas y sistemas de software para que la empresa pueda obtener mejores beneficios [3].

Una cadena de suministro efectiva tiene como propósito encontrar un equilibrio entre la oferta y la demanda mediante la coordinación de todos los procesos internos y externos para garantizar la entrega del producto correcto, al cliente indicado, en el mejor momento y con un costo óptimo. Por lo tanto, la estimación de la demanda futura es una de las tareas cruciales para cualquier organización que cuente con una cadena de suministro debido a que debe tomar decisiones correctas en el momento adecuado para mejorar su competitividad comercial [4].

Un sistema de gestión logística para el manejo de suministros en una cadena de supermercados puede ser una gran ventaja respecto a sus adversarios porque le permite a la empresa tener un mayor control sobre los recursos disponibles para la distribución de sus productos y optimizar los costos que genera este proceso, lo que puede traducirse como mayor beneficio económico.

Se propone crear un sistema que implemente algoritmos genéticos para optimizar las rutas de reparto interno de productos para una cadena de supermercados de Ecuador que cuenta con varias sucursales. El sistema que se desea implementar permitirá automatizar ciertos procesos de gestión logística y tener una mejor planificación de los recursos disponibles.

1.2. Antecedentes y problemática

Uno de los casos de éxito más representativo en el uso de técnicas de inteligencia artificial en procesos de logística fue Deutsche Post DHL, por ser el pionero en el enrutamiento de vehículos usando algoritmos de planificación inteligente. A inicios de siglo, en los años 2000 cuando apenas empezaba el uso del internet para consumo doméstico y de comercio electrónico, esta empresa ya utilizaba sistemas que les ayudaban a planificar las rutas de los repartos de los paquetes [5].

Actualmente, la empresa cuenta con varios centros de investigación para la mejora de sus procesos. Utiliza sistemas de cartografía digital y satelital de las ciudades en las que ofrece sus servicios. También, realiza cruce de información entre distintos mapas, para poder obtener más información como por ejemplo predecir patrones de tráfico, estimar con mejor precisión las entregas, etc [5].

Para el desarrollo de este trabajo se dispone de algunos datos de la empresa Corporación La Favorita, que es una cadena de tiendas especializada en la venta al por menor de diferentes tipos de productos. Cuenta con varias cadenas de supermercados,

jugueterías, ferreterías, tiendas de hogar, etc. Es la empresa privada más grande del país y con un crecimiento a nivel internacional. Tiene las mayores ventas a nivel nacional, ya que supera los 1900 millones de dólares anuales. Cuenta con más de 11.000 trabajadores directos [6].

La Favorita requiere de mucha planificación de logística para un adecuado uso de sus recursos. Dispone de varios supermercados distribuidos a nivel nacional y cuenta con una bodega central desde donde se hace el reparto de los productos hacia el resto de las tiendas. Además, genera un volumen muy grande de ventas por lo que constantemente necesita abastecerse de productos. La empresa cuenta con una flota de vehículos y un grupo de conductores para poder realizar el reparto, pero se debe tener en cuenta que la cantidad disponible es limitada, como también el número de empleados que están asignados para conducirlos.

El problema que se desea resolver es la optimización de múltiples objetivos en la distribución de productos para cada supermercado. Dependiendo la demanda de cada tienda se entrega una cantidad de productos, diariamente se realiza este proceso para evitar que quede desabastecida y no pueda satisfacer las necesidades de sus clientes. El depósito central es donde se almacenan todos los productos para posteriormente ser repartidos hacia los supermercados. La forma de transporte utilizada es terrestre, a través de la red vial nacional lo que permite que todas las tiendas y el depósito central estén conectados por alguna vía. Al problema se debe añadir ciertas restricciones como, por ejemplo: el límite en la capacidad de productos que puede llevar cada camión en cada trayecto, las normativas propias de la empresa y las leyes que están vigentes en el país donde opera la cadena de supermercados.

Los objetivos que se desea minimizar son los siguientes:

1. La distancia recorrida
2. El número de vehículos que se emplea para el reparto.
3. El costo del personal que conduce los camiones.
4. El costo del combustible utilizado para el reparto.
5. El tiempo total que tarda el reparto.

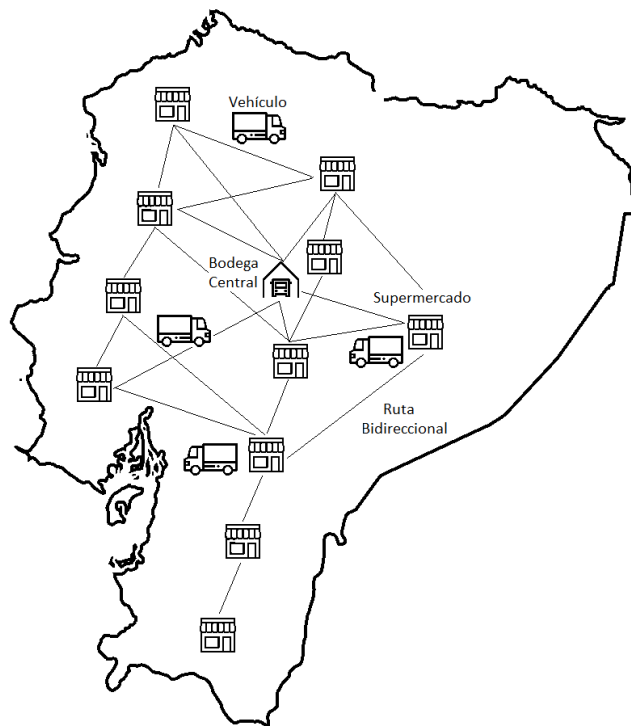


Figura 1.1: Representación del problema que se pretende resolver

1.2.1. Entidades asociadas al problema

Tiendas: Son los supermercados que pertenece a esta cadena. Los atributos de la tienda son: un número de identificación, coordenadas para conocer su ubicación geográfica, una ciudad y una provincia a la que pertenecen y se encuentra físicamente la tienda. Además, cada tienda tiene un requerimiento de productos dependiendo el volumen de ventas que realiza cada día.

Por ejemplo: la tienda número 1 se encuentra en la ciudad *Quito*, en la provincia *Pichincha* y sus coordenadas son: $(-0.1800503, -78.4803628)$

Bodega central: La empresa tiene una sola bodega central en donde se almacenan todos los productos que deben ser repartidos a las diferentes tiendas. Este sitio será el punto de partida de todas las rutas, los vehículos empezarán y finalizarán en este sitio. Además, este será el único lugar donde se cargue los productos a cada vehículo. Los conductores se subirán en los camiones asignados y realizarán los relevos una vez

finalizado el horario de trabajo del anterior conductor. La bodega central también cuenta con una localización mediante coordenadas.

Productos: Son los artículos que deben ser distribuidos. Los artículos están categorizados por grupos de familias a los que pertenecen. tienen un número de registro único con el que pueden ser identificados y están divididos en dos subgrupos *perecibles* y *no perecibles*.

- **Perecibles:** son aquellos productos que requieren refrigeración o que tienen un tiempo de conservación muy corto. Este tipo de productos solo pueden ser transportados por vehículos del tipo frigorífico.
- **No perecibles:** productos que no requieren refrigeración y su tiempo de conservación es prolongado. Solo son transportados por camiones del tipo almacenamiento.

Por ejemplo:

Número	Perecible	Familia
99197	True	FROZEN FOODS
108701	True	SEAFOOD
108833	False	CLEANING
127547	False	GROCERY I
153239	True	MEATS
208386	False	GROCERY II

Ventas: Representan los movimientos ante la demanda de cada tienda por cada producto en una fecha específica. Sus atributos son: la fecha en que se realiza la venta, número de tienda asociada a la venta y la cantidad de unidades vendidas de dicho producto. Las ventas se usan para obtener el listado de productos que van a ser entregados en esa fecha. El listado se usa para la generación de las rutas porque permite conocer los productos que debe cargar cada vehículo cuando visita una tienda para entregar los productos.

Por ejemplo:

	Fecha	Cantidad	Producto	Tienda
1	15-09-2019	5	96995	1
2	15-09-2019	2	103665	1
3	15-09-2019	3	108701	2
4	16-09-2019	7	108831	2
5	16-09-2019	9	108833	7
6	16-09-2019	8	122095	7

Tipos de Vehículos: Son las características que presenta la flota de automotores que tiene disponibles la cadena de supermercados. Un tipo de vehículo está caracterizado por tener: identificador único, nombre, límite de productos que puede almacenar en cada viaje, tipo de combustible que utiliza para movilizarse, total de horas de trabajo disponible para la repartición de los productos, límite de velocidad a la que puede circular el vehículo, número de kilómetros que recorre por cada galón de combustible cargado, además, el tipo de vehículo está dividido en dos subtipos: frigorífico y almacenamiento, para determinar qué tipo de productos puede cargar.

Por ejemplo:

Nombre	Velocidad Máx.	Capacidad	Perecible	Combustible	Km por galón	Máx. T. Trabajo
1 Big Truck	60	5000	False	Diesel	12.5	18
2 Refrigerator Truck	60	5000	True	Extra	11.5	18
3 Medium Truck	70	2500	False	Extra	17.6	18
4 Medium Refrigerator Truck	70	2500	False	Extra	15.5	18
5 Van Truck	80	1250	True	Super	21.3	18

Tipos de combustible: El motor de cada vehículo está fabricado para funcionar con un determinado tipo de combustible que se caracteriza por tener un nombre y un precio de venta por cada galón de combustible.

Por ejemplo: la gasolina Súper cuesta \$2.53, la Extra cuesta \$2.39 y el Diesel cuesta \$1.28

Vehículos: Representan la flota y serán los medios por los que la empresa realizará la entrega de sus productos. Cada vehículo cuenta con un número de identificación, un nombre y tiene asignado un tipo de vehículo al que pertenece.

Por ejemplo:

Nombre	Tipo de vehículo
1 BIG TRUCK 1	Big Truck
61 REFRIGERATOR TRUCK 1	Refrigerator Truck
121 MEDIUM TRUCK 1	Medium Truck
151 MEDIUM REFRIGERATOR TRUCK 1	Medium Refrigerator Truck
181 VAN TRUCK 1	Van Truck

Tipos de permisos de conducir: Existen varios tipos de permisos de conducir según las leyes de tránsito vigentes en Ecuador. Cada permiso autoriza a que una persona pueda conducir uno o varios tipos de vehículos diferentes. Los conductores profesionales tienen la obligación de obtener estos permisos. El chofer podrá operar diferentes tipos según sea su cualificación y su tiempo de experiencia como conductor

profesional. En este estudio cada permiso de conducción tiene un identificador único, un nombre o código y está relacionado con uno o varios tipos de vehículos.

Por ejemplo:

	Nombre	Tipo de vehículo
1	A	Medium Truck
2	B	Medium Truck, Van Truck
3	C	Medium Truck, Medium Refrigerator Truck, Van Truck
4	C1	Big Truck, Refrigerator Truck, Medium Truck, Medium Refrigerator Truck, Van Truck

Conductores: La organización cuenta con un número limitado de conductores que operan la flota de vehículos para que puedan movilizarse. Los conductores tienen un tiempo máximo de horas de trabajo que deben cumplir. Una vez finalizado su tiempo de trabajo realizan relevos para poder descansar. Las rutas son extensas, por lo que se necesitan varios conductores para poder cumplirlas. Las características específicas de los conductores son: nombre, apellido, tipo de permiso de conducir, valor por hora de trabajo de cada conductor y el tipo de permiso de conducir que posee. Además, de las horas normales de trabajo existe la posibilidad de realizar horas extras, aunque están limitadas.

Por ejemplo:

	Nombre	Apellido	Costo Hora	Máx. Trabajo	Máx. Horas Extras	Permiso Conducir
256	Eliot	Perello	15.9	8	4	A
242	María	Ferrer	17.6	8	4	B
241	Lucía	Vilella	20.9	8	4	C
1	Juan José	Ponce	22.4	8	4	C1

Días festivos: Son aquellos días donde generalmente no se trabaja porque se conmemora una fiesta histórica, oficial o eclesiástica. Al tratarse de un negocio tan grande e importante no puede parar sus operaciones, por lo que no todos los trabajadores pueden descansar. Además, durante los días festivos es cuando más ventas se reflejan, por lo que el reparto de productos debe continuar. Para la empresa representa un gasto adicional que sus trabajadores laboren en estas fechas, porque según los reglamentos de trabajo de Ecuador los días festivos o fines de semana que un trabajador labore recibirá pagos adicionales a su sueldo normal.

Carreteras o vías: La red vial del Ecuador es el medio por el que se comunican las tiendas y los vehículos se movilizan. Las vías son de sentido bidireccional, si un vehículo va a visitar una tienda y desea regresar al punto anterior u otro por la

misma ruta puede hacerlo, pero en sentido contrario. Las carreteras o rutas que se usan dependen de las coordenadas de las tiendas.

Configuración general: En la configuración general se pueden establecer las variables que serán de uso general para realizar cálculos y validaciones. Estos parámetros son: el porcentaje adicional por hora extra que se debe pagar al trabajador una vez que ha cumplido su jornada laboral ordinaria, el porcentaje adicional que se paga por hora en el caso de trabajar fin de semana o día festivos. También se configura ciertas normativas de la empresa, por ejemplo: el tiempo máximo de conducción antes de que el conductor deba realizar un descanso para evitar accidentes de tránsito, el tiempo de duración de cada descanso, el tiempo que debe tardar el personal en realizar la carga o descarga de los productos en los vehículos, entre otros.

Por ejemplo:

Tasa hora por festivo o fin de semana	Tasa hora extra	Tiempo de conducción	Tiempo de descanso	Tiempo de carga/descarga
2	1.5	2:00	0:15	0:20

1.3. Objetivos

El objetivo general de este trabajo es diseñar y construir un sistema avanzado de gestión de logística que implemente algoritmos genéticos para la optimización de las rutas de reparto de productos entre las tiendas distribuidas en el territorio ecuatoriano.

Para cumplir el objetivo general se trazaron los siguientes objetivos específicos:

- Analizar el funcionamiento de los algoritmos genéticos y determinar el mejor diseño que se puede implementar para la resolución de problemas de logística.
- Diseñar el algoritmo genético propuesto con las variantes seleccionadas para la resolución del problema de optimización planteado.
- Crear un meta algoritmo y obtener la mejor configuración inicial para este algoritmo y el problema que se desea resolver.
- Realizar evaluaciones con distintas configuraciones y comparar los resultados obtenidos.

- Construir una aplicación web que implemente el algoritmo diseñado y use la mejor configuración encontrada para generar las rutas que los conductores deben seguir para resolver el problema propuesto.

1.4. Estructura del Documento

Este documento se encuentra dividido en seis capítulos, para su mejor comprensión. El primer capítulo contiene toda la introducción al problema. El segundo capítulo profundiza sobre los fundamentos teóricos de los algoritmos genéticos que se van a implementar. El tercer capítulo detalla el diseño del algoritmo genético que se va a usar para resolver el problema. El cuarto capítulo describe la aplicación web creada para la gestión de datos requeridos por el algoritmo para crear las rutas de entrega de productos. En el quinto capítulo presenta todas las evaluaciones realizadas y los resultados obtenidos. Además, se explica el diseño y uso de meta algoritmos para obtener los mejores parámetros para la configuración inicial del algoritmo. También, se realiza un análisis sobre los resultados obtenidos. Por último, están descritas las conclusiones, recomendaciones y trabajo futuro. Al final se encuentra la bibliografía consultada para el desarrollo de este documento.

Capítulo 2

Fundamentos teóricos

La optimización es una rama de las matemáticas, estadísticas y ciencias empíricas que consiste en la selección de las mejores soluciones, con respecto a uno o varios criterios de un conjunto de soluciones factibles. Estos problemas están divididos en continuas y discretas dependiendo el tipo de variables utilizadas. La optimización combinatoria usa variables discretas. [7].

2.1. Problema de ruteo de vehículo (VRP)

Los problemas de ruteo de vehículo son problemas de optimización en la logística del transporte. El objetivo principal de estos problemas es resolver la distribución de productos almacenados en un depósito central hacia un grupo de clientes que poseen una demanda por esos productos, además, se pretende minimizar los costos asociados a las rutas de reparto. Este problema fue presentado por primera vez por George Dantzig y John Ramser. Fue aplicado para resolver un problema relacionado con entrega de combustible [8].

2.1.1. Componentes de problema de ruteo de vehículos

- **Vehículo:** proveedor de servicio que se moviliza a través de la red. Pueden ser de diferentes tipos según el problema: Bicicletas, camiones, autobuses, trenes. Helicópteros, aviones, barcos, etc.
- **Flota de vehículos:** conjunto de vehículos disponibles para el reparto de productos.
- **Sitios por visitar:** lugares o direcciones en los que se encuentran los clientes que van a ser atendidos y tienen una demanda sobre los productos.

- **Red:** vías o carreteras por donde los camiones pueden movilizarse desde cualquier punto (ubicación del cliente o deposito central) hacia otro punto para que pueda cumplir con el reparto de los productos. Generalmente son descritas como grafos.
- **Deposito central:** bodega principal desde donde se despachan los productos, representa una ubicación donde los camiones usualmente están estacionados, recargan y descargan productos, pudiendo ser el punto de partida y el punto final de cada ruta.
- **Conjunto de rutas:** resultado obtenido después de aplicar los algoritmos de optimización, capaz de lograr las demandas de los clientes y minimizar el coste. Los costes de cada ruta pueden ser medidos en distancia, tiempo, dinero, etc.

2.1.2. Funciones objetivo

Varios tipos de funciones objetivos pueden aplicarse a este tipo de problemas:

- Minimizar la distancia total recorrida
- Optimizar el número de camiones utilizados en el reparto
- Optimizar la variación entre el tiempo del viaje y la carga del camión
- Minimizar el número de penalizaciones por servicios de baja calidad (incumplimiento del tiempo de entrega, no se entregan todos los productos, etc.)

2.1.3. Variantes del problema

- **Problema de enrutamiento de vehículo con recogida y entrega:** Cierta grupo de productos necesita ser movido desde ubicaciones de recogida hacia ubicaciones de entrega, debiéndose optimizar las rutas para que la flota de vehículos pueda atender todos los sitios de recogida y lograr entregar todos los productos en sus destinos.
- **Problema de enrutamiento con ventanas de tiempo:** Las visitas a los sitios de entrega de los productos tienen como restricción adicional ventanas de tiempo en las que se debe entregar los productos por lo que deben ser considerados en los procesos.
- **Problema de enrutamiento con capacidad:** Se agrega como restricción la capacidad de productos que puede cargar cada vehículo en cada ruta de entrega.

- **Problema de enrutamiento múltiple:** Los vehículos pueden hacer más de una ruta.
- **Problema de enrutamiento de vehículo abierto:** Cuando los vehículos no están restringidos a terminar en el depósito central.
- **Problema de enrutamiento múltiples depósitos:** Las rutas empiezan y finalizan en diferentes depósitos centrales. Cada depósito estaría restringido con una capacidad de almacenamiento. En este caso cada depósito tendría asignado un costo.

2.1.4. Restricciones adicionales

Varias restricciones pueden ser añadidas según el tipo de problema que se desea modelar.

- Cada vehículo debe tener al menos un conductor para que pueda movilizarse, está restringido por periodos de trabajo durante el día.
- El conductor del vehículo tendrá un número de descansos y un tiempo establecido por descanso.
- Cada vehículo tiene su propia capacidad para todos los productos o para cada tipo de producto.
- El vehículo tiene una velocidad máxima a la que puede circular.
- El uso de cierto tipo de vehículos puede tener un costo asociado.
- La bodega central tiene una capacidad de almacenamiento, puede ser total o por tipo de producto.
- Cuando existan varios depósitos centrales se puede limitar el número de vehículos que están estacionados o que están en la zona de carga y de descarga.
- Los vehículos están restringidos a un horario para carga y descarga de productos.
- Existen horarios de visita para la entrega de productos a los clientes.
- El vehículo puede entregar una o varias veces a un mismo cliente. Depende de la capacidad que tenga cada camión. Las visitas son repetidas hasta que se satisface la demanda del cliente y se entregan los productos que requiere.

2.2. Optimización para problemas multiobjetivo

La mayoría de los problemas de optimización intentan encontrar la mejor solución para un único objetivo, pero también existen aquellos que deben satisfacer múltiples objetivos a la vez. Por ejemplo, los problemas de logística que tienen muchas variables que deben cumplir al mismo tiempo. Estos criterios en muchos casos suelen ser contradictorios.

Para objetivos contradictorios existen varias soluciones que son consideradas como óptimas, ya que cada una satisface a un objetivo concreto. Se debe tener en consideración, que el crecimiento de la satisfacción de un objetivo significa el decrecimiento de otro [9].

Existen técnicas para resolver problemas de optimización multiobjetivo que se han desarrollado en los últimos años. Muchas heurísticas están inspiradas en la naturaleza y utilizan técnicas de inteligencia artificial para simular procesos de evolución biológicos o inteligencia colectiva. Además, existen algoritmos que son exactos o de fuerza bruta que garantizan una solución óptima, pero, representan un mayor costo computacional [9].

Algunos ejemplos de estas técnicas son:

- Algoritmos genéticos
- Algoritmos culturales
- Colonias de hormigas
- Nubes de partículas

2.2.1. Métodos de resolución de problemas multiobjetivo

Agregación de funciones

La aproximación de suma de pesos es una técnica simple para resolución de problemas multiobjetivo. Consiste en combinar todos los objetivos en una única función objetivo para utilizar un solo valor generado. Esta técnica permite usar cualquier algoritmo de optimización de un solo objetivo porque solo se tiene una función de evaluación [10].

Se suele utilizar la función de combinación lineal de los objetivos, utilizando la siguiente ecuación:

$$F = \sum_{i=1}^k w_i f_i(x) \quad (2.1)$$

donde cada w_i son los pesos que se le da a cada función objetivo, los pesos deben ser normalizados:

$$F = \sum_{i=1}^k w_i = 1 \quad (2.2)$$

Selección del vector de pesos El vector de pesos se utiliza para asignar la preferencia sobre cada objetivo en la resolución del problema. Matemáticamente, una solución obtenida con el mismo peso para cada objetivo suele dar menos conflictos. Además, esta técnica puede ser aplicada calculando cada valor objetivo en una función individual. Los resultados son seleccionados en un vector de pesos acorde con las preferencias y los resultados obtenidos de cada función individual.

Ventajas La técnica es fácil de implementar y permite escoger un objetivo para controlar el resto usando los resultados de la función individual. La solución óptima normalmente suele ser una solución Pareto óptima.

Desventajas Es muy sensible a la especificación de los pesos y solo se obtiene una solución en cada iteración del algoritmo.

2.3. Algoritmos Genéticos

Los algoritmos genéticos son heurísticas de búsqueda aplicadas a problemas de optimización. Están inspirados en la biología, específicamente en la teoría de la evolución propuesta por Charles Darwin en su libro “El Origen de las Especies” [11].

Los principios básicos de los algoritmos genéticos fueron establecidos por Holland, y se encuentran bien descritos en varios textos [12][13][14][15][16].

El algoritmo genético clásico se basa en un grupo de soluciones (individuos) que representan un problema a resolver. Una solución es un candidato potencial para un óptimo del problema. Su representación juega un papel importante, ya que determina la elección de los operadores genéticos. Los individuos evolucionan a lo largo de las generaciones, siguiendo los principios de selección natural. La selección imita el proceso natural en que el más fuerte sobrevive.

La variación, imita el proceso evolutivo para la creación de nuevas formas de vida utilizando el operador de cruce y de mutación.

Para obtener resultados favorables utilizando este algoritmo hay que tomar en cuenta

que es muy importante la codificación del problema y la elección de una adecuada función objetivo [17].

Algoritmo 1: Algoritmo genético

Result: Solución óptima encontrada

Genera la población inicial;

while *condición de parada* == *Falso* **do**

 Selección de individuos;

while *Número de iteraciones de evolución* **do**

 Cruzar dos individuos seleccionados para obtener descendientes;

 Mutar descendientes;

 Insertar descendientes en la población;

 Elimina individuos restantes;

end

end

2.3.1. Codificación

Los individuos son posibles soluciones al problema que pueden estar representados por un conjunto de parámetros o *genes*, cuando se encuentran agrupados forman una ristra de valores. Los *cromosomas* son el conjunto de genes que representan las características del individuo. Generalmente, se usa el 0 y 1 para representar el alfabeto de los individuos, debido a que la mayoría de la teoría que fundamenta los algoritmos genéticos lo utiliza para que sea más fácil su explicación, usualmente tiene representaciones más complejas.

Genotipo: Conjunto de genes que existen en el núcleo celular de cada individuo; contiene la información necesaria para la construcción del organismo.

Fenotipo: Conjunto de caracteres visibles que un individuo presenta como resultado de la interacción entre su genotipo y el medio. Es un conjunto de soluciones que se obtienen a partir de la función de evaluación.

2.3.2. Población

Es el conjunto de individuos que tiene un tamaño máximo. El límite debe ser establecido en los parámetros iniciales del algoritmo, el tamaño de la población es muy importante ya que una población pequeña corre el riesgo de no cubrir adecuadamente el espacio de búsqueda, mientras que si la población tiene un tamaño excesivamente

grande puede causar problemas de costo computacional. La población suele inicializarse generando individuos al azar. Cada gen de cada individuo debe contener un valor del posible alfabeto con una probabilidad uniforme. Se han realizado algunos estudios de la inicialización de la población utilizando otras heurísticas, en muchos casos puede ayudar a que el tiempo de convergencia disminuya, pero también se corre el riesgo de estancarse en óptimos locales [11].

2.3.3. Función de evaluación

La función de evaluación está diseñada para cada problema específico. El valor de aptitud mide la calidad de las soluciones encontradas por el algoritmo, es evaluado respecto a la solución objetivo que se desea alcanzar. El diseño de la función objetivo es parte fundamental en todo el algoritmo genético, debido a que las soluciones se van a guiar en este paso.

El valor de aptitud debe ser lo más cercano a la realidad para construir una buena función objetivo, porque en muchos de los problemas con muchas restricciones no se encuentra soluciones válidas.

Para los individuos que no cumplen con las restricciones del problema se puede realizar algunas acciones:

- No considerarlo para la población y seguir iterando en cruces y mutaciones hasta conseguir una solución válida.
- Marcar su valor de aptitud en cero para que la probabilidad de selección sea nula.
- Reconstruir el individuo que no cumple con las restricciones.
- Penalizar la solución para que en la selección se tome en cuenta el número de restricciones que no se cumplen.

El número de llamadas a la función de aptitud tiene que ser reducido porque el rendimiento de un algoritmo genético para resolver un problema generalmente se mide en número de evaluaciones realizadas hasta encontrar una solución óptima. Hay problemas en los que ejecutar la función de aptitud es costoso y puede requerir mucho tiempo en realizar la evaluación.

2.3.4. Restricciones

Las restricciones reducen la cantidad de soluciones que puede generar el algoritmo. Generalmente, un problema no encuentra solución cuando muchas restricciones son

aplicadas. El resultado de una función de restricción puede ser del tipo booleano, indicando la factibilidad del problema. Existen varias estrategias que se pueden emplear para intentar superar las restricciones de un problema de optimización [11].

Reparación

Las reparaciones permiten que las soluciones inalcanzables se vuelvan alcanzables. En la reparación se puede remover material genético duplicado o también añadir material que falte para completar la solución.

Las soluciones continuas son reparadas proyectando un individuo inviable sobre uno que sea viable. Una solución viable es la más cercana o más adecuada al óptimo de todo el conjunto de la población y sirve como punto de referencia para decidir qué se debe agregar o quitar para que una solución sea alcanzable [11].

2.3.5. Selección

El operador de selección es utilizado para permitir la convergencia hacia soluciones óptimas, consiste en seleccionar las mejores soluciones para que sean los padres de nuevos individuos descendentes. Los individuos descendentes son remplazados con los peores individuos de la población. Una nueva generación de individuos es creada a partir de los mejores resultados generados por la población inicial. Los mejores individuos de cada generación son seleccionados para conservar el avance hacia la solución óptima. La selección se realiza con los resultados obtenidos de la función de evaluación, los menores o mayores valores son seleccionados cuando el problema de optimización intenta minimizar o maximizar los objetivos.

Muchos algoritmos de selección se basan en la aleatoriedad, los padres son seleccionados al azar con una distribución uniforme. La probabilidad de ser seleccionado depende del valor de aptitud obtenido. Los resultados son normalizados para que todas las soluciones tengan una probabilidad positiva de ser seleccionadas. La aleatoriedad permite que las mejores soluciones puedan ser olvidadas, este método se utiliza para superar los óptimos locales.

El torneo es otra forma para seleccionar los mejores resultados, consiste en elegir aleatoriamente un número de soluciones que son ordenados según el valor de aptitud. Este mecanismo de selección puede usarse para decidir qué padres van a realizar el cruce. El proceso se lo conoce como *supervivencia*, porque solo las mejores soluciones se conservan, este principio se basa en los estudios realizados por Darwin [11].

2.3.6. Cruce

El cruce es un operador que permite la combinación de material genético entre dos o más soluciones. Dos soluciones suelen ser usados para realizar el proceso de cruce debido a que este algoritmo está inspirado en la naturaleza, pero hay ciertos problemas en los que el cruce se lo realiza con más de dos soluciones.

Este operador implementa un mecanismo para mezclar todo el material genérico y de esta forma generar una nueva posible solución. El cruce puede ser implementado de diversas formas, varía según la configuración de la proporción de mezcla de cada padre. Por ejemplo: dividir en partes iguales el aporte de los dos padres, solicitar que un padre aporte mayor cantidad de material genético, incluso hacer una selección aleatoria de material genético de ambos padres [18].

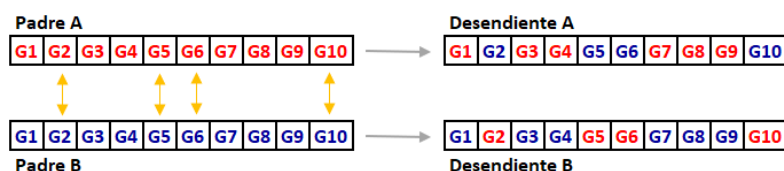


Figura 2.1: Representación del cruce entre dos individuos padres

2.3.7. Mutación

El segundo operador del algoritmo genético es la mutación. Este operador consiste en cambiar ciertos genes del individuo aleatoriamente. La tasa de mutación es configurada en los parámetros iniciales del algoritmo y es usada para calcular el porcentaje de cambio aplicado.

Tres requerimientos deben cumplirse para realizar la mutación.

- El primer requerimiento es la accesibilidad, cada punto de la solución debe ser accesible desde un punto arbitrario del espacio de la solución. A medida que aumentan las restricciones es más difícil cumplir este punto.
- El segundo requerimiento es la imparcialidad, la mutación no debe enfocarse en una dirección de búsqueda específica.
- El tercer requisito es la escalabilidad, debe ofrecer un grado de libertad para ser adaptable.



Figura 2.2: Representación de mutación de un individuo

2.3.8. Reducción

El operador de reducción se usa para conservar el tamaño de la población establecido en la configuración inicial del algoritmo. La población suele incrementar en el proceso evolutivo donde se añaden nuevos individuos. Existen dos técnicas que permiten realizar este proceso. La primera, es la *reducción elitista*, consiste en ordenar la población según su valor de aptitud obtenido en la evaluación y elimina los individuos menos favorecidos. La segunda técnica, consiste en que los hijos descendientes reemplazan a los padres por lo que después del cruce y/o la mutación son generados el mismo número de descendientes que de padres [11].

2.3.9. Finalización

La condición de finalización determina cuando el bucle de evolución va a terminar. El número iteraciones en que va a realizar todo el proceso evolutivo (generación, evaluación de la función objetivo y selección de los mejores individuos) es definido en la configuración inicial del algoritmo.

Otra condición útil para la terminación del bucle es que si en un determinado número de generaciones el valor de aptitud no presenta mejoras significativas se termina el proceso evolutivo [11].

2.3.10. Experimentación

La experimentación es muy importante porque permite probar diferentes configuraciones y versiones del algoritmo para obtener los mejores resultados. Los algoritmos genéticos tienen un comportamiento no determinista y requieren al menos realizar veinticinco experimentos con cada configuración para que los resultados puedan ser comparables [11].

2.3.11. Multimodal

Los multimodales existen cuando varios óptimos locales y globales causan dificultad porque el algoritmo no sabe si ha llegado a un óptimo local o global. Un nicho se crea cuando solo existe una población de soluciones candidatas, la respuesta suele quedar atrapada en áreas de óptimos locales. El tamaño de la población debe aumentar

para deshacer el estancamiento en nichos, porque una mayor cantidad de soluciones aumenta la posibilidad de saltar a otro nicho [11].

Estancamiento prematuro Es producido en un espacio de soluciones de muy baja probabilidad. Las soluciones de baja probabilidad son causadas por la disminución de las tasas de mutación debido a que existen muy pocas opciones de solución. Varias estrategias pueden ser empleadas para evitar este problema. Por ejemplo: un operador de mutación que sea capaz de adaptarse a la forma de las restricciones en donde la tasa de mutación sea mucho más baja cuando detecte que hay baja probabilidad de solución [11].

Varias estrategias se pueden aplicar para solucionar los problemas de multimodalidad:

Reinicio

El reinicio consiste en repetir la búsqueda múltiples veces, cada vez el algoritmo arranca de un lugar en el espacio de soluciones elegido aleatoriamente. La estrategia es aplicada cambiando los parámetros del algoritmo en cada ejecución, el cambio se usa para explorar nuevas áreas en el espacio de búsqueda. Los parámetros que pueden ser modificados son: el tamaño de la población, la tasa de mutación y de cruce, la estrategia de selección de individuos, etc. El mismo óptimo local es encontrado múltiples veces cuando es necesario una población más grande o una tasa de mutación más alta.

Otro enfoque del reinicio consiste en la búsqueda global en el espacio de los óptimos locales. La tasa de mutación global puede ser controlada si se junta la búsqueda local con un mecanismo de mutación que actúa a nivel global. La tasa de mutación global es aumentada si el mismo óptimo local es repetido varias veces, pero, si nuevas soluciones son encontradas después de cada aproximación local disminuye la tasa. El último resultado del óptimo local es utilizado como base para la tasa de mutación global en la nueva ejecución del algoritmo [11].

Búsqueda de novedades

Esta técnica tiene como objetivo explorar las áreas desconocidas del espacio de soluciones. La búsqueda de novedades requiere una evaluación de la novedad y la singularidad de la solución. Existen varias medidas para evaluar, por ejemplo: la distancia entre una solución y una población de soluciones o la distancia de una solución a un histórico de soluciones recorridas. La distancia euclidiana puede ser usada como medida.

Este proceso puede ejecutarse después de inicializar la población o después de la mutación y cruce. Una novedad existe si una solución es distinta a las de la población inicial. Esta técnica puede ser empleada como medición del error para los algoritmos meta genéticos porque si se encuentra alguna novedad significa que existen mejores parámetros para el algoritmo interno [11].

2.3.12. Selección de parámetros

El éxito de un algoritmo genético está ligado a la selección de los parámetros. Existen dos tipos de parámetros, exógenos y endógenos. Los parámetros exógenos son globales aplicados para todo el algoritmo, por ejemplo: el tamaño de la población o el número de iteraciones en que el proceso evolutivo es repetido. Mientras que, los parámetros endógenos son aquellos que están a nivel de cromosoma. Estos parámetros son usados para auto regular el funcionamiento.

Muchas configuraciones estáticas fueron propuestas para el ajuste de parámetros, pero se ha descubierto que no todas las configuraciones funcionan igual para cada problema. El ajuste de los parámetros es resuelto como un problema de optimización. Varios métodos pueden ser aplicados para el ajuste de los parámetros [19].

Tasa de mutación automática Acoplar una tasa de mutación al contador de generaciones y utilizar un factor de multiplicación para que en cada generación disminuya cierto porcentaje de la tasa de mutación. Otra variante de esta técnica es aplicar un factor reductor de valor fijo en cada generación. El uso de estos factores depende del problema y de los parámetros aplicados. La reducción permite la convergencia hacia un óptimo. Las tasas de mutación pueden tener un control evolutivo de adaptación automática no supervisado. Para el proceso de mutación en la búsqueda de tasas hay operadores especiales que permiten una aproximación logarítmica lineal del óptimo [19].

$$\sigma t = \sigma^{\tau \cdot \mathcal{N}(0, 1)} \quad (2.3)$$

Donde τ es la tasa de mutación de las tasas de mutación.

Meta Algoritmos Genéticos Los meta algoritmos genéticos son otra estrategia para conseguir mejores configuraciones iniciales. Puede emplearse algoritmos genéticos convencionales, pero el problema de optimización a resolver es ajustar los parámetros de otro algoritmo genético interno que va a resolver un problema de optimización real. El valor de aptitud es calculado con el resultado obtenido de las ejecuciones repetidas del algoritmo genético interno. La ejecución debe repetirse varias veces porque su resultado es estocástico. Varias medidas deben ser usadas para decidir si los resultados obtenidos por la función de adecuación del algoritmo genético externo son

razonables. Puede usarse: la mediana, la media o la mejor ejecución. También, la desviación estándar debe tenerse en consideración para asegurarse de que un valor atípico no sea responsable de la evaluación del conjunto de parámetros actual. Esta estrategia suele encontrar mejores parámetros comparado con los que se obtienen configurando manualmente el algoritmo. La búsqueda de estos parámetros es costosa, debido a que para cada evaluación del algoritmo externo se debe realizar múltiples ejecuciones de algoritmo interno [20].

Decodificadores Los decodificadores se utilizan para aprender metamodelos en espacios restringidos. A un espacio restringido se le asigna uno que tiene mejores condiciones para ser solucionado y menos restricciones. La ejecución del meta algoritmo genético es realizada en un espacio sin muchas restricciones, lo que permite que el proceso de optimización sea más fácil. Además, permite aprender un metamodelo con restricción lineal, que después puede ser utilizado para solucionar problemas con más restricciones [20].

Capítulo 3

Construcción del algoritmo genético

Lo que se plantea resolver es un problema de optimización combinatorio complejo y pertenece a la clase *NP-Hard* debido a su naturaleza no se permite utilizar métodos exactos para resolverlo. Los algoritmos genéticos (GA) proporcionan una técnica de inteligencia artificial para la búsqueda de una solución aproximada a los problemas de optimización complejos. A comparación de otras heurísticas existentes, GA no es muy difícil de implementar y se puede adaptar a muchos problemas y tipos de restricciones. Además, se ha utilizado en muchos estudios para resolver problemas similares obteniendo resultados muy satisfactorios [21] [22] [23].

3.1. Diseño del algoritmo en base al problema

3.1.1. Configuración inicial

Los algoritmos genéticos requieren de una configuración inicial antes de ser ejecutados porque necesitan ciertos parámetros que controlan el funcionamiento del algoritmo. La heurística diseñada requiere ciertas configuraciones adicionales a las de un algoritmo genético común porque requiere más valores que van a usarse en la función de evaluación de los objetivos.

La configuración inicial del algoritmo diseñado es la siguiente:

- **Fecha de ejecución:** Es la fecha para la que se va a generar las rutas. Se usa para consultar la proyección de ventas y determinar si ese día es festivo o fin de semana.

- **Tamaño de la población:** El número de individuos que tendrá la población inicial.
- **Número de evoluciones:** La cantidad de veces que se repetirá el proceso evolutivo del algoritmo genético.
- **Tasa de mutación:** Porcentaje de individuos que serán mutados.
- **Tasa de cruce:** Porcentaje de individuos padres que realizarán cruce para la generación de descendientes.
- **Número de fallos permitidos:** Valor usado como condición de parada para determinar si el problema tiene solución.
- **Peso del objetivo de distancia:** Peso otorgado al objetivo de distancia en la función de evaluación.
- **Peso del objetivo de número de vehículos:** Peso otorgado al objetivo de número de vehículos en la función de evaluación.
- **Peso del objetivo de costo de personal:** Peso otorgado al objetivo de costo de personal en la función de evaluación.
- **Peso del objetivo de costo de combustible:** Peso otorgado al objetivo de costo de combustible en la función de evaluación.
- **Peso del objetivo de tiempo de reparto:** Peso otorgado al objetivo de tiempo de reparto en la función de evaluación.

Normalización de los pesos

Los pesos deben ser normalizados antes de la ejecución del algoritmo porque se necesita ajustar los valores para que la suma de todos sea uno.

Configuración inicial interna

El algoritmo requiere otros valores adicionales a los de la configuración inicial, estos datos se obtienen de base de datos y fuentes externas.

- **Tiendas visitadas por fecha:** Listado de las tiendas que recibirán productos en la fecha ingresada en la configuración inicial. Un listado de productos es generado filtrando la proyección de las ventas por la fecha ingresada. Para completar la información del listado, por cada tienda se consulta las coordenadas de localización y el nombre de identificación.

- **Bodega central:** Tienda donde se almacenan los productos y se carga o descarga la mercadería en los camiones. Es consultada de la base de datos para que pueda ser usada en la ejecución del algoritmo.
- **Configuración general:** Valores usados en el cálculo de las funciones objetivo y para la validación de las restricciones.
- **Vehículos** Los vehículos son consultados antes de que inicie el algoritmo para evitar múltiples consultas en la base de datos y mejorar el rendimiento de la ejecución. Se usan para generar individuos.
- **Conductores** Los conductores son consultados antes de que inicie el algoritmo para evitar múltiples consultas en la base de datos y mejorar el rendimiento de la ejecución. Se usan para generar individuos.

3.1.2. Gen

Los algoritmos genéticos están inspirados en la naturaleza, específicamente en las cadenas de ADN. Un cromosoma está dividido en genes, que almacena cada uno de los rasgos de los seres vivos. Los genes tienen una serie de atributos que representan parte de la solución del problema.

El gen empleado para este algoritmo está compuesto de una ruta, un vehículo, un grupo de conductores asignados e información adicional. Entre la información adicional se encuentran los siguientes atributos: la distancia total recorrida, el tiempo total de la ruta, el listado de los productos que se debe entregar, la capacidad máxima disponible por cada trayecto, la velocidad a la que circula en camión y un nombre para que pueda ser identificado.

Ruta

La ruta es el listado de todas las tiendas que debe visitar el camión en la entrega de productos, incluida la bodega central, porque ese es el lugar de abastecimiento de productos, también porque ahí inicia y finaliza la ruta. Las tiendas visitadas pueden estar repetidas porque significa que se repartió varias veces a una misma tienda.

Listado de productos

El listado de productos es la guía para saber qué productos deben entregarse en cada supermercado. También, determina la cantidad despachada de cada artículo. Los productos de este listado deben ser del mismo tipo que el del vehículo asignado (perecibles o no perecibles).

Conductores

El listado de los conductores es asignado a cada gen aleatoriamente, teniendo en cuenta el tipo de licencia que disponen y el vehículo que ha sido asignado. El tiempo de trabajo de los conductores debe ser igual al tiempo que el vehículo dispone para la entrega.

Vehículo

El vehículo es asignado al gen aleatoriamente del grupo de automotores que se encuentran registrados en la base de datos. Solo se asigna un automotor por gen.

Otros atributos

- El nombre asignado a cada gen es único y sigue una secuencia. Sirve para que pueda ser identificado.
- La distancia total es la suma de todos los kilómetros que recorre el vehículo para cumplir con la ruta.
- El límite de velocidad es un atributo copiado del tipo de vehículo asignado al gen.
- El tiempo máximo de trabajo del gen, es el tiempo máximo que puede estar asignado el camión para el reparto.
- El tiempo total de la entrega es la suma del tiempo que se demora el automotor en recorrer la ruta sumado los descansos y los tiempos de carga y descarga.
- La capacidad disponible es un atributo copiado del tipo de vehículo.

Gen 1
Nombre: Ruta 1
Vehículo: Van Truck 1
Conductores: [Mireya Torres, Luis Yugsi]
Ítems: [{ítem: 295923, Tienda: 43, Cantidad: 2}; {ítem: 87537, Tienda: 19, Cantidad: 17}]
Tiendas Visitadas: [Bodega Central, Tienda 19, Bodega Central, Tienda 19, Bodega Central, Tienda 19, Tienda 43, Bodega Central]
Distancia: 19
Límite de Velocidad: 90
Tiempo máximo de Trabajo: 18
Tiempo total de entrega: 13
Capacidad: 5

Figura 3.1: Ejemplo de gen

3.1.3. Cromosoma

El cromosoma que va a ser implementado para este algoritmo consiste en un conjunto de genes que fueron mencionados en la sección anterior, agrupados para formar una posible solución para el problema. El cromosoma tendrá una dimensión variable debido a la naturaleza del problema. En la creación del cromosoma, para cada gen se selecciona vehículos y conductores aleatoriamente, no todos los vehículos pueden ser usados para cada solución debido a que existe un número limitado de conductores que pueden conducir cada tipo de vehículo.

El cromosoma se considera válido cuando cumple las siguientes condiciones:

- **Vehículos** Los vehículos no se pueden repetir, cada gen debe tener asignado uno.
- **Conductores** Los conductores no se pueden repetir y deben tener un tipo de permiso de conducción válido para operar el tipo de vehículo asignado al gen. El tiempo de trabajo de los conductores no puede superar el tiempo máximo de trabajo del gen.
- **Lista Productos** Un gen debe tener una lista de productos con elementos del mismo tipo al vehículo asignado. La lista de tiendas visitadas debe ser igual a la que consta en la lista de productos que debe entregar.
- **Ruta** La ruta empieza y finaliza en la bodega central. Puede retornar muchas veces a la bodega central para abastecer de productos el camión para cumplir con la entrega. Una misma tienda también puede ser visitada varias veces por el mismo automotor.
- **Tiempo total de entrega** El tiempo de la entrega no puede superar al tiempo de trabajo del gen.
- **Nombre del gen** El nombre del gen debe ser único entre todo el cromosoma.

Cromosoma				
Gen 1	Gen 2	Gen 3	Gen 4	Gen 5
Nombre: Ruta 1 Vehículo: Van Truck 1 Conductores: [Mireya Torres, Luis Yuga] Ítems: [{Ítem: 295923, Tienda: 43, Cantidad: 2}, {Ítem: 87537, Tienda: 19, Cantidad: 17}] Tiendas Visitadas: [Bodega Central, Tienda 19, Bodega Central, Tienda 19, Bodega Central, Tienda 19, Tienda 43, Bodega Central] Distancia: 19 Límite de Velocidad: 90 Tiempo máximo de Trabajo: 18 Tiempo total de entrega: 13 Capacidad: 5	Nombre: Ruta 2 Vehículo: Big Truck 27 Conductores: [Pedro González, Juan Jose Ponce] Ítems: [{Ítem: 146734, Tienda: 16, Cantidad: 5}, {Ítem: 632531, Tienda: 18, Cantidad: 4}, {Ítem: 20934, Tienda: 2, Cantidad: 6}] Tiendas Visitadas: [Bodega Central, Tienda 16, Tienda 18, Tienda 2, Bodega Central, Tienda 2, Bodega Central] Distancia: 9 Límite de Velocidad: 50 Tiempo máximo de Trabajo: 18 Tiempo total de entrega: 14.9 Capacidad: 10	Nombre: Ruta 3 Vehículo: Refrigerator Truck 16 Conductores: [Paulo Cisneros, Camila Méndez] Ítems: [{Ítem: 52052, Tienda: 31, Cantidad: 7}, {Ítem: 96536, Tienda: 16, Cantidad: 14}] Tiendas Visitadas: [Bodega Central, Tienda 31, Tienda 16 Bodega Central, Tienda 16, Bodega Central, Tienda 16, Bodega Central] Distancia: 21 Límite de Velocidad: 50 Tiempo máximo de Trabajo: 18 Tiempo total de entrega: 15.6 Capacidad: 10	Nombre: Ruta 4 Vehículo: Van Refrigerator Truck 2 Conductores: [Marcos Torres, Lorena Chicaiza] Ítems: [{Ítem: 49946, Tienda: 1, Cantidad: 5}, {Ítem: 87537, Tienda: 1, Cantidad: 11}, {Ítem: 87537, Tienda: 45, Cantidad: 2}] Tiendas Visitadas: [Bodega Central, Tienda 1, Bodega Central, Tienda 1, Bodega Central, Tienda 45, Bodega Central] Distancia: 12.5 Límite de Velocidad: 90 Tiempo máximo de Trabajo: 18 Tiempo total de entrega: 16.6 Capacidad: 5	Nombre: Ruta 5 Vehículo: Medium Truck 5 Conductores: [Gladis Loachamin, Paul Sierra] Ítems: [{Ítem: 51462, Tienda: 48, Cantidad: 2}, {Ítem: 64532, Tienda: 4, Cantidad: 7}] Tiendas Visitadas: [Bodega Central, Tienda 4, Bodega Central, Tienda 48, Bodega Central] Distancia: 25.6 Límite de Velocidad: 60 Tiempo máximo de Trabajo: 18 Tiempo total de entrega: 13.5 Capacidad: 7

Figura 3.2: Ejemplo de cromosoma

Creación de cromosoma

Para crear un cromosoma que pueda ser usado por el algoritmo diseñado se debe seguir los siguientes pasos:

1. Crear un arreglo del tamaño de todos los vehículos disponibles
2. Instanciar un gen por cada posición del arreglo.
3. Cada gen debe tener asignado un vehículo aleatoriamente, teniendo en cuenta que no se puede repetir.
4. Asignar al gen los atributos adicionales usando el tipo de vehículo asignado, la capacidad por trayecto, el límite de velocidad, el tiempo máximo de trabajo.
5. Asignar un nombre único para cada gen.
6. Para la asignación de cada gen se escogerá en forma aleatoria los conductores que estén disponibles y con permiso de conducir para el tipo de vehículo del gen
7. Eliminar los genes que no se asignó conductores o vehículos.

3.1.4. Individuo

El individuo representa una posible solución para el problema. Está compuesto de un cromosoma válido y un valor de aptitud, que se obtiene después de realizar la evaluación en la función objetivo.

Nombre individuo	Cromosoma	Valor de aptitud
Individuo 1	Gen 1	7527.62
	Gen 2	
	Gen 3	
	Gen 4	
	Gen 5	

Figura 3.3: Ejemplo de individuo

3.1.5. Población

La población es un arreglo de individuos que son utilizados por el algoritmo genético para la búsqueda de una solución óptima. La longitud del arreglo está establecida en la configuración inicial antes de la ejecución del algoritmo.

Nombre individuo	Cromosoma	Valor de aptitud
Individuo 1	Gen 1	7527.62
	Gen 2	
	Gen 3	
	Gen 4	
	Gen 5	
Individuo 2	Gen 1	13941.34
	Gen 2	
	Gen 3	
	Gen 4	
Individuo 3	Gen 1	8566.73
	Gen 2	
	Gen 3	
	Gen 4	
	Gen 5	
	Gen 6	
Individuo 4	Gen 1	17736.88
	Gen 2	
	Gen 3	
Individuo 5	Gen 1	15670.95
	Gen 2	
	Gen 3	
	Gen 4	
	Gen 5	
	Gen 6	

Figura 3.4: Ejemplo de población

Creación de la población

Para inicializar la población primero se creará un arreglo del tamaño que fue ingresado en la configuración inicial. Por cada posición del arreglo se creará un individuo con un cromosoma válido. Los requisitos serán validados antes de insertar el individuo a la población. En caso de que la validación falle se generará otro individuo.

Verificación de restricciones

Una vez que se ha generado el cromosoma se debe validar las siguientes restricciones, en caso de que no cumpla será eliminado el gen que incumple las normas.

- Todos los genes deben tener al menos un conductor asignado y no se puede repetir dentro del cromosoma.
- Todos los genes deben solo tener un vehículo asignado y no se puede repetir
- La suma de las horas de trabajo de los conductores no puede ser mayor a las del tiempo máximo de trabajo del gen.

3.2. Creación de una solución

A continuación, se detalla los pasos necesarios para crear una solución válida

1. La población inicial con los cromosomas de cada individuo ya deben haberse iniciado.
2. Por cada individuo
3. Seleccionar un gen y una tienda aleatoriamente.
4. Calcular la cantidad total de productos que la tienda requiere. Si el resultado es igual a cero, se elimina la tienda para que no vuelva a ser seleccionada.
5. Calcular la cantidad total de productos del mismo subtipo del vehículo que pertenece al gen (perecibles o no perecibles). Si es igual a cero se vuelve a escoger otro gen.
6. Verificar el tiempo que el gen está asignado a la entrega de productos. Si el resultado es menor al tiempo máximo de trabajo, continua. Caso contrario, se selecciona otro gen.

Para calcular el tiempo de trabajo del gen:

Para obtener el tiempo total de reparto, se multiplica la velocidad máxima asignada al gen y la distancia total recorrida.

Para calcular el número de descansos, dividir el tiempo total recorrido para el tiempo de conducción permitido (configuración general).

Para calcular el tiempo de descanso, multiplicar el número de descansos con el tiempo de duración de cada descanso (Configuración general).

Para calcular el tiempo de carga y descarga de productos, multiplicar el número de tiendas visitadas por el tiempo de carga y descarga de productos (Configuración general).

Sumar *tiempo de conducción* más *tiempo de descanso* más *tiempo de carga y descarga*.

7. Verificar la capacidad disponible del vehículo. Si es mayor a cero, se agrega la tienda al listado de visitadas y se continua.
8. Si el total de productos requeridos por la tienda no sobrepasa la capacidad del gen se asignan todos los productos al listado al gen. Caso contrario; se recorrerá todo el listado de productos y se añadirá solo aquellos que están dentro de la capacidad de almacenamiento disponible.

9. Una vez añadidos los productos al gen, son eliminados del listado de productos de la tienda para que no exista duplicidad de entregas.
10. Si la capacidad disponible del gen llega a cero, se reinicia y se añade al listado de tiendas visitadas la bodega central. Porque al no tener más espacio para la entrega de productos el camión debe regresar para cargar nuevos artículos y continuar con el reparto.
11. Existe una condición que verifica que si una tienda se ha seleccionado varias veces y no reduce su total de productos durante número de veces se considera que el cromosoma no tiene solución porque no se encuentra un gen para satisfacer la demanda de esa tienda.

Se crea un nuevo individuo y repite todo el proceso.

12. Realizar la limpieza del cromosoma consiste en eliminar aquellos genes que no visitaron ninguna tienda.
13. Se verifica que todos los genes finalicen en la bodega central, caso contrario se agrega la bodega central a la ruta.
14. Finalmente, si todo ha ido bien se retorna el cromosoma válido.

3.3. Evaluación del individuo

La técnica de agregación de funciones va a ser utilizada para evaluar todos los objetivos que fueron propuestos para optimizar. Para cada objetivo se asigna un peso para determinar qué objetivo es más importante satisfacer, porque existen objetivos que son conflictivos y se debe dar prioridad según lo que se desea optimizar. El resultado de la función de evaluación es la combinación de todos los objetivos sumando los resultados obtenidos en la evaluación de cada función objetivo multiplicado por el peso correspondiente. El resultado es un solo valor numérico.

$$\begin{aligned}
 fitness = & distanceRate * distanceResult + numVehiclesRate * numVehiclesResult \\
 & + fuelCostRate * fuelCostResult + deliveryTimeRate * deliveryTimeResult \\
 & + staffCostRate * staffCostResult
 \end{aligned}$$

3.3.1. Evaluación de la distancia total

Para la evaluación de este objetivo se va a usar la fórmula de Haversine, que permite el cálculo de la distancia entre dos puntos de un globo, sabiendo su longitud

y latitud. Se recorre todas las tiendas que se han visitado en la ruta, incluida la bodega central del inicio. La distancia total del gen se obtiene con la suma de todas las distancias calculadas entre las tiendas de la ruta. [24].

$$\begin{aligned} lat1 &= \text{radians}(pos1.x) & lon1 &= \text{radians}(pos1.y) \\ lat2 &= \text{radians}(pos2.x) & lon2 &= \text{radians}(pos2.y) \\ a &= \sin\left(\frac{lat2 - lat1}{2}\right)^2 + \cos(lat1) * \cos(lat2) * \sin\left(\frac{lon2 - lon1}{2}\right)^2 \\ c &= 2 * \arctan(\sqrt{a}, \sqrt{1 - a}) \\ distance &= 6373,0 * c \end{aligned}$$

La distancia total del cromosoma se suma los resultados de cada gen.

$$totalDistance = \sum_{i=0}^N i.distance$$

3.3.2. Evaluación del número de vehículos utilizados

El cálculo de este objetivo se calcula contando el número de genes que tiene el cromosoma, porque cada uno tiene asignado un solo vehículo. Se debe considerar que las rutas que no visitaron ninguna tienda fueron eliminadas y solo quedaron aquellas en las que realmente se hace entrega de productos.

$$numTrucks = \sum_{i=0}^N 1$$

3.3.3. Evaluación del coste total de combustible

La evaluación del coste de combustible de cada gen se calcula de la siguiente manera:

1. Calcular el número de galones necesarios para cumplir toda la ruta, dividir la distancia total para el número de kilómetros que puede recorrer el tipo de vehículo asignado al gen.
2. Calcular el coste de combustible por gen, multiplicar el número de galones por el precio de combustible asociado al tipo de vehículo asignado al gen.

La suma de todos los costes de combustible de cada gen da como resultado el coste total de combustible para todo el cromosoma.

$$totalGallons = \sum_{i=0}^N \frac{distance}{kmPerGallon}$$

$$totalFuelCost = totalGallons * fuelCost$$

3.3.4. Evaluación del tiempo de reparto

La evaluación del tiempo de reparto se evalúa con el tiempo máximo entre los genes del cromosoma, debido a que todos los automóviles trabajan simultáneamente y el mayor tiempo abarca al resto de tiempos de otros genes.

Tiempo de entrega por ruta El tiempo de entrega de los productos para cada gen es calculado de la siguiente manera:

1. Calcular la distancia total del gen.
2. Para obtener el tiempo total de reparto, se multiplica la velocidad máxima asignada al gen y la distancia total recorrida.

$$drivingTime = \frac{distance}{maxSpeedLimit}$$

3. Para calcular el número de descansos, dividir el tiempo total recorrido para el tiempo de conducción permitido (configuración general).

$$numBreaks = \frac{drivingTime}{maxDrivingTime}$$

4. Para calcular el tiempo total de descanso, multiplicar el número de descansos con el tiempo de duración de cada descanso (Configuración general).

$$breakTime = numBreaks * breakTime$$

5. Para calcular el tiempo de carga y descarga de productos, multiplicar el número de tiendas visitadas por el tiempo de carga y descarga de productos (Configuración general).

$$totalLoadTime = numVisitedStores * loadTime$$

6. Sumar *tiempo de conducción* más *tiempo de descanso* más *tiempo de carga y descarga*.

$$totalDeliveryTime = drivingTime + breakTime + totalLoadTime$$

3.3.5. Evaluación del coste del personal involucrado

La evaluación del coste de personal se debe determinar el tiempo de trabajo de cada conductor y después multiplicar por el valor de hora. Dependiendo si es feriado o fin de semana se multiplica por la tasa correspondiente. En el caso de que el tiempo de trabajo se exceda se debe calcular las horas extras y multiplicar por la tasa correspondiente.

El cálculo de las horas de trabajo de cada conductor se evalúa cada vez que el vehículo regresó a la bodega central. Si el tiempo es igual o mayor al máximo de trabajo del conductor significa que terminó su horario. En el caso de que sea el último conductor el tiempo de trabajo finalizará cuando acabe la ruta.

- Cada vez que el vehículo retorna a la bodega central calcular el tiempo del trayecto.
- Si el tiempo total del trayecto es mayor o igual al tiempo de trabajo del conductor o si es el último conductor y la última tienda visitada
- Para calcular el costo de cada conductor, multiplicar el precio por hora del conductor por el número total de horas de trabajo multiplicado por la tasa correspondiente si ese día es festivos o fines de semana.

$$staffCost = driverHourCost * actualWorkTime * weekendHolidayRate$$

- Para el caso de las horas extras, se debe restar el tiempo total de trabajo - el tiempo de trabajo del conductor, la diferencia son las horas extras trabajadas. Serán pagadas multiplicando el precio por hora del conductor por la tasa de horas extras.

$$normalCost = driverHourCost * driverMaxWorkTime$$

$$remainingTime = actualWorkTime - driverMaxWorkTime$$

$$extraCost = driverHourCost * remainingTime * extraRate)$$

$$staffCost = normalCost + extraCost$$

Para obtener el valor total por cromosoma se debe sumar el resultado obtenido en cada gen.

3.4. Selección de los individuos padres

Para realizar la selección de los individuos padres para usar el operador de cruce se tiene que seguir los siguientes pasos:

1. Evaluar toda la población.
2. Ordenar la población por el valor de aptitud de cada individuo de forma ascendente para que los valores más bajos tengan mayor probabilidad de ser seleccionados. La forma de ordenar es ascendente porque se desea minimizar los objetivos.
3. Se crea un torneo con los diez mejores individuos, para garantizar que la descendencia va a mejorar los resultados obtenidos.
4. Elegir los dos individuos padres de forma aleatoria.

Población		Población Ordenada		Torneo	
Nombre Individuo	Valor Aptitud	Nombre Individuo	Valor Aptitud	Nombre Individuo	Valor Aptitud
Individuo 1	7527.62	Individuo 17	7405.19	Individuo 17	7405.19
Individuo 2	13941.34	Individuo 18	7466.91	Individuo 18	7466.91
Individuo 3	8566.73	Individuo 1	7527.62	Individuo 1	7527.62
Individuo 4	17736.88	Individuo 3	8566.73	Individuo 3	8566.73
Individuo 5	15670.95	Individuo 7	9517.59	Individuo 7	9517.59
Individuo 6	14185.14	Individuo 9	11382.84	Individuo 9	11382.84
Individuo 7	9517.59	Individuo 16	11449.03	Individuo 16	11449.03
Individuo 8	11681.55	Individuo 8	11681.55	Individuo 8	11681.55
Individuo 9	11382.84	Individuo 19	12547.35	Individuo 19	12547.35
Individuo 10	12995.77	Individuo 20	12776.57	Individuo 20	12776.57
Individuo 11	13829.66	Individuo 10	12995.77		
Individuo 12	13800.62	Individuo 12	13800.62		
Individuo 13	14977.25	Individuo 11	13829.66		
Individuo 14	16169.17	Individuo 2	13941.34		
Individuo 15	15609.22	Individuo 6	14185.14		
Individuo 16	11449.03	Individuo 13	14977.25		
Individuo 17	7405.19	Individuo 15	15609.22		
Individuo 18	7466.91	Individuo 5	15670.95		
Individuo 19	12547.35	Individuo 14	16169.17		
Individuo 20	12776.57	Individuo 4	17736.88		

Figura 3.5: Ejemplo del proceso de selección de los individuos

3.5. Operador de cruce

Los cromosomas son asimétricos debido a que no tienen la misma longitud porque muchos genes son eliminados del cromosoma debido a que no satisfacen las restricciones. La cantidad de genes que se realiza el cruce es controlada por la tasa de cruce ingresado en la configuración inicial.

El cruce solo se realiza sobre los automóviles y los conductores asignados al gen porque el listado de productos que está asignado a cada gen no se puede alterar para evitar inconsistencias en los productos que recibe cada tienda.

Los cambios alteran los resultados significativamente, porque el cambio de automóvil varía la capacidad del gen y produce que la distancia sea alterada debido a que cambia el número de trayectos que se realiza el vehículo para el reparto. Otro cambio efectuado cuando el vehículo cambia es el tiempo total de reparto, debido a que no todos los automóviles se mueven a la misma velocidad. El costo de personal varía porque cada gen asigna conductores con diferentes costos por hora.

Para hacer el cruce entre dos individuos se debe seguir los siguientes pasos:

1. Seleccionar un gen aleatoriamente del cromosoma A
2. Seleccionar un gen aleatoriamente del cromosoma B
3. Usando un valor aleatorio y la tasa de cruce se decide si se realiza el cruce entre los dos genes.
4. Obtener listado de vehículos asignados al cromosoma A
5. Obtener listado de vehículos asignados al cromosoma B
6. Obtener listado de conductores asignados al cromosoma A
7. Obtener listado de conductores asignados al cromosoma B
8. Si el subtipo de vehículo del gen A es el mismo del gen B y Si el vehículo del gen B no está asignado en cromosoma A
 - Asignar el vehículo del gen B al vehículo del gen A Caso contrario.
 - Asignar un vehículo disponible en el cromosoma A reemplazando el vehículo del gen A.
9. Repetir paso 8 pero de gen A hacia gen B.

10. Si conductores gen B no están asignados en cromosoma A.
 Asignar conductores gen B hacia el gen A. Caso contrario.
 Asignar el mismo número de conductores, que estén disponibles en el cromosoma A hacia el gen A.
11. Repetir paso 10 pero de gen A hacia B.
12. Si la capacidad del nuevo vehículo del gen A o B es diferente a la anterior.
 Reparar el gen.
 Si la reparación del gen A o B dio error no se realiza ningún cambio sobre los genes seleccionados.

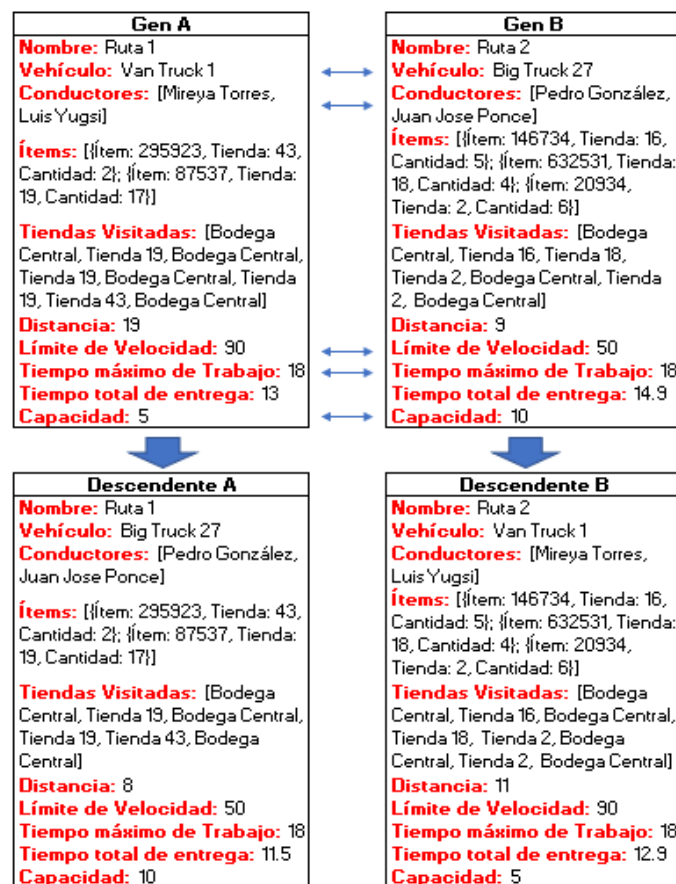


Figura 3.6: Ejemplo de cruce

3.6. Operador de mutación

El operador de mutación consiste en que los genes del cromosoma cambian sus valores de forma aleatoria. Es generalmente aplicado sobre los individuos descendentes, pero se controla con la tasa de mutación ingresado en la configuración inicial. El operador de mutación usado en este algoritmo genético solo va a cambiar el vehículo y los conductores asociados al gen.

Para realizar el proceso de mutación se debe seguir los siguientes pasos:

1. Para cada individuo descendente
2. Seleccionar aleatoriamente un gen del cromosoma.
3. Usando un valor aleatorio y la tasa de mutación se decide si el gen va a ser mutado.
4. Consultar todos los vehículos disponibles que son del mismo subtipo del gen seleccionado.
5. Elegir un vehículo aleatoriamente.
6. Reemplazar el vehículo actual del gen con el vehículo seleccionado.
7. Consultar todos los conductores que tienen permiso para conducir el nuevo tipo de vehículo asignado al gen y que se encuentran disponibles.
8. Reemplazar los conductores seleccionando aleatoriamente del listado teniendo en cuenta las restricciones asociadas al tiempo máximo de trabajo.
9. Si la capacidad del nuevo vehículo cambió.

Reparar el gen.

Si la reparación del gen dio error no se realiza ningún cambio sobre el gen seleccionado.

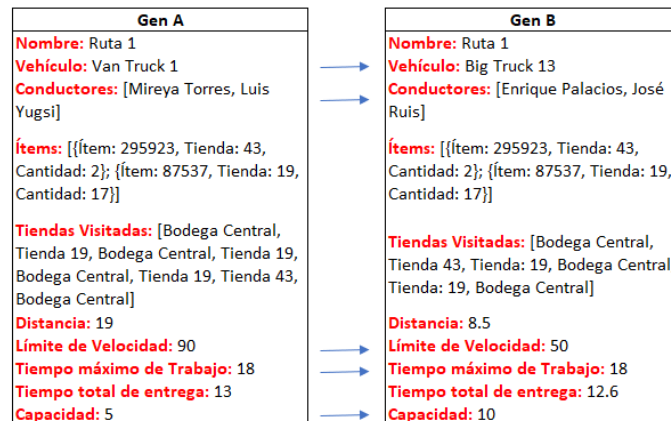


Figura 3.7: Ejemplo de mutación

3.7. Reparar el gen

El proceso de reparación de un gen sirve para que cuando este se daña o se vuelve inconsistente y necesita ser arreglado para que no afecte a la solución. Usualmente un gen se daña cuando se ha sometido al operador de cruce o de mutación.

Para reparar un gen se debe seguir el siguiente proceso:

1. Crear un nuevo gen con el mismo nombre, vehículo, conductores asignados, capacidad y tiempo máximo de trabajo.
2. Extraer el listado de tiendas visitadas y agrupar los artículos entregados a cada tienda en esa ruta.
3. Inicializar los otros atributos del gen: la ruta solo se queda con la bodega central, la distancia recorrida y el tiempo de reparto regresan a cero.
4. Escoger una tienda aleatoriamente del listado extraído.
5. La cantidad total de productos es calculada para validar si se puede entregar los productos en un solo viaje o en varios. En caso de que sea en varios viajes se agregará la tienda y la bodega central a la ruta por las veces que tenga que retornar el vehículo para cumplir con el reparto. En el caso de que solo fuera un viaje se inserta la tienda a la ruta.
6. Repetir pasos 4 y 5 hasta que la cantidad total de productos a entregar en todas las tiendas sea cero.

7. Al finalizar se calcular la distancia total y el tiempo de reparto para validar si aún cumple con las restricciones.
8. Si el gen sigue satisfaciendo las restricciones se reemplaza por gen dañado. Caso contrario:
9. Se retorna un error

3.8. Reducción de población

La eliminación de los individuos se realiza usando el valor de aptitud. Después de cada iteración en el proceso evolutivo los descendientes son evaluados y son añadidos a la población. La población es ordenada de forma ascendente para que los menores valores sean los primeros y los últimos individuos que están fuera del tamaño de la población son descartados. Siempre se mantiene el mismo tamaño inicial de la población.

Población		Población Ordenada		Eliminación de individuos	
Nombre Individuo	Valor Aptitud	Nombre Individuo	Valor Aptitud	Nombre Individuo	Valor Aptitud
Individuo 1	7527.62	Individuo 17	7405.19	Individuo 17	7405.19
Individuo 2	13941.34	Individuo 18	7466.91	Individuo 18	7466.91
Individuo 3	8566.73	Individuo 1	7527.62	Individuo 1	7527.62
Individuo 4	17736.88	Individuo 3	8566.73	Individuo 3	8566.73
Individuo 5	15670.95	Individuo 23	9111.54	Individuo 23	9111.54
Individuo 6	14185.14	Individuo 7	9517.59	Individuo 7	9517.59
Individuo 7	9517.59	Individuo 22	11139.57	Individuo 22	11139.57
Individuo 8	11681.55	Individuo 9	11382.84	Individuo 9	11382.84
Individuo 9	11382.84	Individuo 16	11449.03	Individuo 16	11449.03
Individuo 10	12995.77	Individuo 8	11681.55	Individuo 8	11681.55
Individuo 11	13829.66	Individuo 21	12167.60	Individuo 21	12167.60
Individuo 12	13800.62	Individuo 19	12547.35	Individuo 19	12547.35
Individuo 13	14977.25	Individuo 20	12776.57	Individuo 20	12776.57
Individuo 14	16169.17	Individuo 10	12995.77	Individuo 10	12995.77
Individuo 15	15609.22	Individuo 12	13800.62	Individuo 12	13800.62
Individuo 16	11449.03	Individuo 11	13829.66	Individuo 11	13829.66
Individuo 17	7405.19	Individuo 2	13941.34	Individuo 2	13941.34
Individuo 18	7466.91	Individuo 6	14185.14	Individuo 6	14185.14
Individuo 19	12547.35	Individuo 13	14977.25	Individuo 13	14977.25
Individuo 20	12776.57	Individuo 15	15609.22	Individuo 15	15609.22
Individuo 21	12167.60	Individuo 5	15670.95		
Individuo 22	11139.57	Individuo 14	16169.17		
Individuo 23	9111.54	Individuo 4	17736.88		

Tamaño de la población: 20

Figura 3.8: Ejemplo de eliminar individuos

3.9. Condición de parada

La condición de parada es usada para determinar cuando finaliza la ejecución del algoritmo.

1. El algoritmo tiene que terminar cuando todo el proceso de inicialización de la población se ha generado correctamente, se ha cumplido el número de iteraciones de evolución entre los cromosomas y se obtuvo una solución válida para el problema. La solución que devuelve el algoritmo para este problema es la de menor valor de aptitud, porque se desea minimizar los resultados.
2. Cuando se intenta generar un individuo durante varias repeticiones que está controlado por un contador que incrementa cuando se obtiene un individuo que no es válido. Si el valor del contador es igual al valor del parámetro *Número de errores permitidos* se retorna un mensaje de error indicando que no existe solución para el problema.
3. Si en el proceso evolutivo al realizar mutación o cruce se obtiene una cantidad de individuos sin solución que son controlados por un contador que se incrementa cuando no se puede reparar una solución. Si el valor del contador es igual al valor de la variable de *Número de errores permitidos* se retorna un mensaje de error indicando que no existe solución para el problema.
4. Cuando los tiempos máximos de trabajo de cada gen han alcanzado los valores permitidos y aún existen tiendas pendientes, significa que no existe solución para el problema.

Capítulo 4

Aplicación web para la gestión de rutas

En este capítulo se presentará una aplicación web que fue creada especialmente para resolver el problema de optimización en la gestión logística de supermercados. La aplicación funciona como un gestor de los datos necesarios para la ejecución del algoritmo que genera las rutas.

En la aplicación está implementado el algoritmo genético que fue diseñado para resolver el problema planteado. La aplicación está pensada para que los resultados puedan ser interpretados de una manera fácil y organizada para que puedan ser usada por los conductores y los coordinadores de logística sin mucho esfuerzo y para que la entrega de los diferentes productos en grandes volúmenes sea una tarea más eficiente y sencilla. Se pretende que la aplicación sea usada diariamente para obtener las rutas de cada día ya que los productos deben ser entregados con esa frecuencia.

4.1. Bases de datos

La base de datos utilizada en la elaboración de este proyecto se basa en una competición realizada por la empresa Corporación La favorita en la plataforma “Kaggle”, que es un sitio enfocado a competiciones de *data science* e *inteligencia artificial*.

La empresa proporcionó estos datos para realizar el torneo que consistía en generar pronóstico de ventas para el año dos mil diez y siete. El rango de la información proporcionada para la competición es desde el año dos mil doce hasta agosto del dos mil diez y siete. Los datos de esta competición se basan en información real, pero usando máscaras y con ciertas restricciones para evitar fuga de información sensible.

Gran cantidad de los datos y la estructura en la que se encuentran organizados son útiles para poder resolver el problema planteado. Existen muchas columnas en las tablas que repiten muchos valores por lo que se ha decidido normalizar las tablas para una mejor compresión [25].

Las tablas disponibles en este conjunto de datos son:

- **Holiday_events:** Tabla que contiene el registro de todos los días festivos hasta la fecha de corte. Las columnas son: fecha del día festivo, tipo de festivo, tipo de localización, localización, descripción y un campo lógico que indica si el día fue transferido.
- **Ítems:** Tabla que almacena un listado de los productos disponibles. Las columnas son: número de ítem, familia a la que pertenece, clase y un campo lógico que indica si es perecible.
- **Oil:** Ecuador es un país muy dependiente del precio del petróleo por lo que en la mayoría de los negocios tiene bastante influencia. En esta tabla se almacena el precio diario de barril de petróleo. Las columnas son: fecha y precio
- **Sample_submission:** Tabla que usa para el envío de resultados en la competición de kaggle. Las columnas son: ventas Unitarias y fecha.
- **Stores:** Tabla que contiene un registro de todas las tiendas disponibles en las diferentes provincias de Ecuador. Las columnas son: número de tienda, ciudad, provincia, tipo y clúster.
- **Test:** Esta tabla almacena las ventas del mes de agosto del año dos mil diez y siete. Fue utilizada para probar los modelos desarrollados en la competición. Las columnas son: fecha, número de tienda, número de ítem y un campo lógico que indica si el artículo se encuentra en promoción.
- **Train:** Esta tabla contiene las ventas que se realizó durante cada día en cada supermercado. Las columnas son: fecha, número de tienda, número de ítem, número de unidades vendidas y un campo lógico que indica si el artículo se encontraba en promoción.
- **Transactions:** Esta tabla contiene un registro de las transacciones diarias realizadas por cada tienda. Las columnas son: fecha, número de tienda y número de transacciones.

Es importante destacar que las tablas Train y Test no pudieron ser insertadas en la base de datos debido a la gran cantidad de datos que tienen. Por lo tanto, se usó la herramienta Pandas para poder cargar estos datos en la aplicación y que puedan ser utilizados.

4.1.1. Bases de datos adicionales

El algoritmo genético requiere de información adicional para que pueda validar las restricciones del problema. Estructuras propias fueron creadas para que almacenen todos los datos adicionales.

- **Vehicle_Type:** Esta tabla almacena la información sobre los diferentes tipos de vehículos que puede tener el supermercado. Las columnas son: nombre, capacidad, tipo de combustible, tiempo máximo de trabajo permitido, límite de velocidad máximo, kilómetros por galón y un campo lógico que indica si es del tipo frigorífico o almacenamiento.
- **Vehicle:** Esta tabla es utilizada para guardar los vehículos que el supermercado tiene disponibles. Las columnas son: número de registro y tipo de vehículo
- **Fuel_Type:** Esta tabla almacena los diferentes tipos de combustible que pueden utilizar los automotores. Las columnas son: nombre y precio
- **Driver_License:** Esta tabla almacena el listado de permisos de conducir profesionales asociadas a los conductores. Las columnas son; nombre y varios tipos de vehículos asociados
- **Driver:** Esta tabla almacena la información de los conductores que realizan los repartos de los productos en las tiendas. Las columnas son: nombre, apellido, tipo de permiso de conducir, precio por hora de trabajo, número de horas de trabajo normal y número de horas adicionales.
- **Configuration:** Esta tabla guarda información adicional que requiere el algoritmo para hacer validaciones de las restricciones y para realizar cálculos de las funciones de evaluación. Las columnas son: tasa de tiempo extra, tasa de fin de semana, tiempo de conducción máximo, duración del descanso y duración del tiempo de carga o descarga.
- **Log:** Esta tabla almacena toda la información relacionada con la ejecución del algoritmo. Las columnas son: fecha de ruta, tamaño de la población, número de iteraciones, tasa de mutación, tasa de cruce, número máximo de errores

permitidos, peso de distancia, peso de costo de personal, peso de costo de combustible, peso de tiempo de entrega, resultado de distancia, resultado de número de vehículos, resultado de costo de personal, resultado de costo de combustible, resultado de tiempo de entrega. resultado de aptitud, errores generados durante la ejecución, un campo lógico que indica si fue seleccionada como la mejor ejecución, campo lógico que indica si la ejecución del algoritmo fue completada, tiempo total de ejecución en segundos y fecha de creación del registro.

- **MetaLog:** Esta tabla acumula toda la información relacionada con la ejecución del meta algoritmo que permite obtener la mejor configuración para el algoritmo interno que se encarga de la búsqueda de las mejores rutas, las columnas son: fecha de ruta, tamaño de la población, número de iteraciones, tasa de mutación, tasa de cruce, número máximo de errores permitidos, resultado de la mejor aptitud obtenida, errores durante la ejecución del meta algoritmo, tiempo total y fecha de creación.

4.1.2. Fuentes de datos adicionales

Para generar las estructuras mencionadas en la sección anterior hizo falta la consulta de información externa para que puedan ser rellenadas. A continuación, se describirá la fuente de donde se obtuvieron estos valores:

- En la tabla de días festivos se agregó las fechas festivas desde la fecha de corte hasta el final del año 2019. Los datos fueron obtenidos de documentos publicados por el ministerio de turismo de Ecuador [26].
- Las coordenadas de las tiendas no estaban incluidas en los datos originales de la competición, fueron obtenidas usando la dirección de cada tienda usando el sistema de Google Maps [27].
- Los datos de las ventas de este año se obtuvieron a partir de las ventas que se tiene en los datos, se realizó el cálculo de la variación de ventas por cada producto, tienda, fecha y año. Las ventas fueron proyectadas para el año actual.
- No se puede obtener información de los tipos de vehículos que dispone la empresa así que se ha decidió crear cinco tipos básicos de vehículos: camiones grandes, frigoríficos grandes, camiones medianos, camiones pequeños, frigoríficos pequeños.
- La cantidad de vehículos que dispone la empresa es un dato que no se pudo obtener así que se han creado vehículos aleatoriamente, los suficientes para poder satisfacer la demanda.

- En Ecuador se comercializan para el sector privado principalmente tres tipos de combustible, Diesel, Extra y Súper. Cada una es diferente en su nivel de pureza y octanaje, por lo que depende de cada vehículo para que pueda ser utilizada. Los precios que se utilizan en la aplicación son los publicados por el organismo encargado de regularlos [28].
- Los permisos de conducir utilizados en el sistema, están basados en los publicados por la Agencia Nacional de Tránsito del Ecuador con ligeras modificaciones para el uso de la aplicación [29].
- El listado de los conductores que operan los vehículos fue generado aleatoriamente, porque esos datos son privados y la empresa no los publica.
- La configuración son valores que ya depende del negocio, fueron seleccionados para cumplir con este requisito.

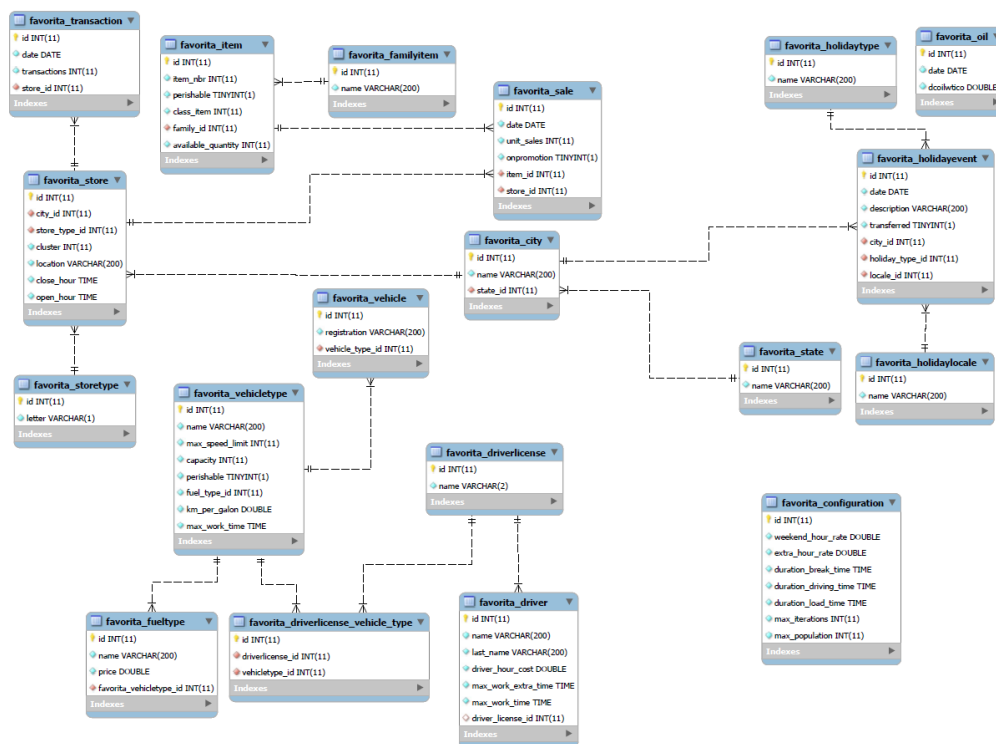


Figura 4.1: Modelo de la base de datos

4.2. Tecnologías utilizadas

Como servidor de base de datos SQL se eligió MySQL porque es muy rápido, multiproceso, multiusuario y robusto. MySQL Server está diseñado para sistemas de producción de carga pesada y de misión crítica, así como para integrarse en software implementado en masa.[30]

El lenguaje en el que fue desarrollada la aplicación es Python, porque es un lenguaje de programación orientado a objetos claro y potente, comparable a Perl, Ruby, Scheme o Java, utiliza una sintaxis elegante, lo que hace que los programas escritos en este lenguaje sean más fáciles de leer. Es fácil de usar porque simplifica el funcionamiento de su programa. Python es ideal para el desarrollo de prototipos y otras tareas de programación ad-hoc, sin comprometer la mantenibilidad. Viene con una gran biblioteca estándar que admite muchas tareas de programación comunes, como conectarse a servidores web, buscar texto con expresiones regulares, leer y modificar archivos [31].

Como marco de trabajo para el desarrollo web se eligió Django porque está escrito en Python de alto nivel que fomenta el desarrollo rápido y el diseño limpio y pragmático. Creado por desarrolladores experimentados, se ocupa de gran parte del desarrollo web, por lo que permite la concentración del tiempo en escribir la aplicación sin necesidad de preocuparse por el desarrollo web. Es gratis y de código abierto.[32]

4.3. Funcionamiento de la aplicación web

La aplicación funciona como un gestor de información que permite el ingreso y la consulta de datos, está compuesta por varias pantallas para la manipulación de la información que se encuentra almacenada en la base de datos. La aplicación debe ser manejada por los coordinadores logísticos porque ellos conocen la información que debe ser ingresada y el funcionamiento de la empresa.

Para acceder a la aplicación web se debe utilizar un explorador web, porque la arquitectura empleada es de cliente servidor. Es decir que se encuentra alojada en un servidor y para que los usuarios puedan acceder deben conectarse a internet o a una red privada.

4.3.1. Pantalla de inicio

La pantalla inicial es la primera página que se carga cuando un usuario abre la aplicación. En la parte izquierda de la pantalla se puede observar un panel con todas las opciones por las que puede navegar. En la parte central se puede observar un gráfico con la estimación de ventas por cada mes para este año.

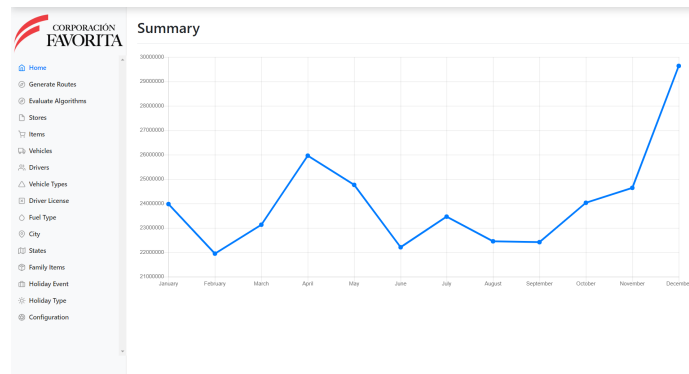


Figura 4.2: Pantalla de inicio de la aplicación web

4.3.2. Tiendas

En esta pantalla se encuentra el mantenimiento de las tiendas a las que se puede entregar productos. La página permite consultar, crear, modificar o eliminar tiendas. Para acceder a esta página hay que seleccionar la opción *Stores* en el menú izquierdo, cuando se abre la pantalla muestra el listado de todas las tiendas disponibles. Al hacer doble clic sobre cualquier registro se mostrará un formulario con el detalle de los datos y además se permite editar. En el caso de agregar una nueva tienda hay que presionar sobre el botón *Add*. Para eliminar una tienda, presionar sobre el botón en forma de bote de basura junto a cada registro.

Id	City	Type	Cluster	Location	
1	Quito	D	13	(-0.180003; -78.4831628)	
2	Quito	D	13	(-0.184177; -78.4929478)	
3	Quito	D	8	(-0.2588474; -78.5250348)	
4	Quito	D	9	(-0.2859509; -78.5454689)	
5	Santo Domingo	D	4	(-0.2523222; -78.1678399)	
6	Quito	D	13	(-0.300887; -78.4321729)	
7	Quito	D	8	(-0.1979681; -78.4410295)	
8	Quito	D	8	(-0.2066792; -78.4893973)	
9	Quito	B	6	(-0.1966806; -78.5088161)	
10	Quito	C	15	(-0.0919246; -78.4766938)	

Figura 4.3: Pantalla de tiendas en aplicación web

4.3.3. Productos

En esta pantalla se encuentra el mantenimiento de los productos que están disponibles para el reparto. La página permite consultar, crear, modificar o eliminar productos. Para acceder a esta página hay que seleccionar la opción *Items* en el menú izquierdo, cuando se abre la pantalla muestra el listado de todos los productos disponibles. Al hacer doble clic sobre cualquiera de los registros se mostrará un formulario con el detalle de los datos y además se permite editar. En el caso de agregar un nuevo producto hay que presionar sobre el botón *Add*. Para eliminar un producto, presionar sobre el botón en forma de bote de basura junto a cada registro.

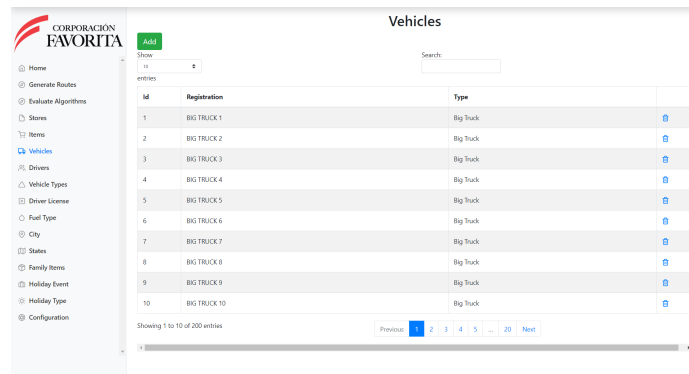
Id	Item #	Family	Class	Perishable	
1	96995	GROCERY I	1093	False	
2	99197	GROCERY I	1067	False	
3	102001	CLEANING	3008	False	
4	103120	GROCERY I	1028	False	
5	103865	BREAD/BAKERY	2712	True	
6	105374	GROCERY I	1045	False	
7	105375	GROCERY I	1045	False	
8	105516	GROCERY I	1045	False	
9	105577	GROCERY I	1045	False	
10	105689	GROCERY I	1034	False	
11	105737	GROCERY I	1044	False	
12	105857	GROCERY I	1052	False	
13	106716	GROCERY I	1032	False	
14	108079	GROCERY I	1030	False	
15	108634	GROCERY I	1075	False	

Figura 4.4: Pantalla de productos en aplicación web

4.3.4. Vehículos

En esta pantalla se encuentra el mantenimiento de los vehículos que están disponibles para el reparto de productos. La página permite consultar, crear, modificar o

eliminar vehículos. Para acceder a esta página hay que seleccionar la opción *Vehículos* en el menú izquierdo, cuando se abre la pantalla muestra el listado de todos los vehículos disponibles. Al hacer doble clic sobre cualquiera de los registros se mostrará un formulario con el detalle de los datos y además se permite editar. En el caso de agregar un nuevo vehículo hay que presionar sobre el botón *Add*. Para eliminar un vehículo, presionar sobre el botón en forma de bote de basura junto a cada registro.



The screenshot shows a web application interface for managing vehicles. The page title is 'Vehicles'. On the left is a sidebar menu with options like Home, Generate Routes, Evaluate Algorithms, Stores, Items, Vehicles (selected), Drivers, Vehicle Types, Driver License, Fuel Type, City, States, Family Items, Holiday Event, Holiday Type, and Configuration. The main content area has a search bar and a table with 10 rows of vehicle data. Each row has a trash icon for deletion. The table columns are 'Id', 'Registration', and 'Type'. The data is as follows:

Id	Registration	Type
1	BIG TRUCK 1	Big Truck
2	BIG TRUCK 2	Big Truck
3	BIG TRUCK 3	Big Truck
4	BIG TRUCK 4	Big Truck
5	BIG TRUCK 5	Big Truck
6	BIG TRUCK 6	Big Truck
7	BIG TRUCK 7	Big Truck
8	BIG TRUCK 8	Big Truck
9	BIG TRUCK 9	Big Truck
10	BIG TRUCK 10	Big Truck

At the bottom of the table, it says 'Showing 1 to 10 of 200 entries'. There are pagination controls for 'Previous', '2', '3', '4', '5', '20', and 'Next'.

Figura 4.5: Pantalla de vehículos en aplicación web

4.3.5. Conductores

En esta pantalla se encuentra el mantenimiento de los conductores que transportan los productos en los vehículos para el reparto. La página permite consultar, crear, modificar o eliminar conductores. Para acceder a esta página hay que seleccionar la opción *Drivers* en el menú izquierdo, cuando se abre la pantalla muestra el listado de todos los conductores disponibles. Al hacer doble clic sobre cualquiera de los registros se mostrará un formulario con el detalle de los datos y además se permite editar. En el caso de agregar un nuevo conductor hay que presionar sobre el botón *Add*. Para eliminar un conductor, presionar sobre el botón en forma de bote de basura junto a cada registro.

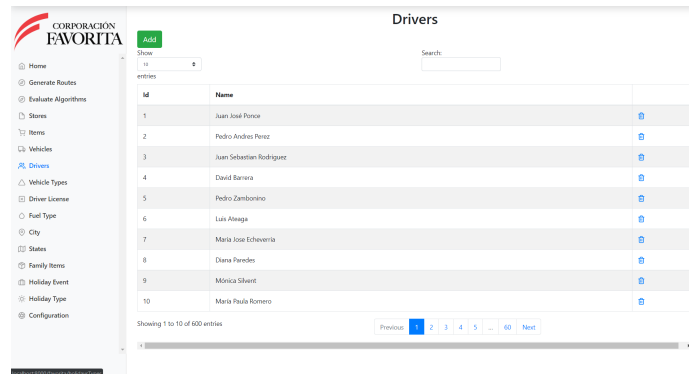


Figura 4.6: Pantalla de conductores en aplicación web

4.3.6. Tipo de Vehículos

En esta pantalla se encuentra el mantenimiento de los tipos de vehículos asociados a cada vehículo. La página permite consultar, crear, modificar o eliminar tipos de vehículos. Para acceder a esta página hay que seleccionar la opción *Vehicles Types* en el menú izquierdo, cuando se abre la pantalla muestra el listado de todos los tipos de vehículos disponibles. Al hacer doble clic sobre cualquiera de los registros se mostrará un formulario con el detalle de los datos y además se permite editar. En el caso de agregar un nuevo tipo de vehículo hay que presionar sobre el botón *Add*. Para eliminar un tipo de vehículo, presionar sobre el botón en forma de bote de basura junto a cada registro.

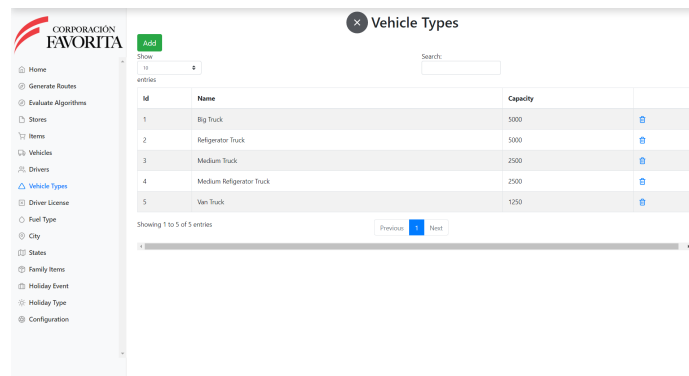
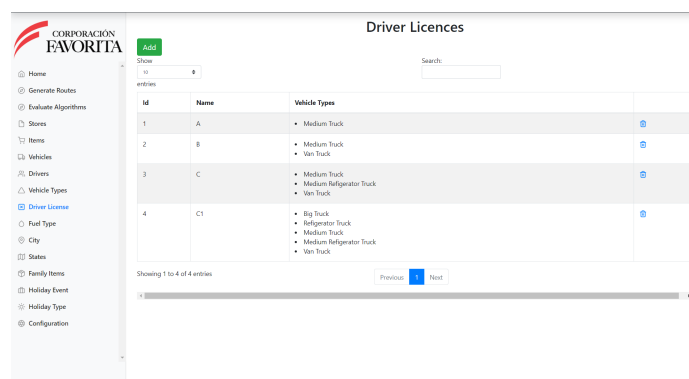


Figura 4.7: Pantalla de tipos de vehículos en aplicación web

4.3.7. Permisos de conducir

En esta pantalla se encuentra el mantenimiento de los permisos de conducir que puede tener un conductor. La página permite consultar, crear, modificar o eliminar permisos de conducir. Para acceder a esta página hay que seleccionar la opción *Driver Licenses* en el menú izquierdo, cuando se abre la pantalla muestra el listado de todos los permisos de conducir disponibles. Al hacer doble clic sobre cualquiera de los registros se mostrará un formulario con el detalle de los datos y además se permite editar. En el caso de agregar un nuevo permiso de conducir hay que presionar sobre el botón *Add*. Para eliminar un permiso de conducir, presionar sobre el botón en forma de bote de basura junto a cada registro.



Id	Name	Vehicle Types
1	A	• Medium Truck
2	B	• Medium Truck • Van Truck
3	C	• Medium Truck • Medium Refrigerator Truck • Van Truck
4	C1	• Big Truck • Refrigerator Truck • Medium Truck • Medium Refrigerator Truck • Van Truck

Figura 4.8: Pantalla de permisos de conducir en aplicación web

4.3.8. Tipo de combustible

En esta pantalla se encuentra el mantenimiento de los tipos de combustible asociados a los diferentes tipos de vehículos. La página permite consultar, crear, modificar o eliminar tipos de combustible. Para acceder a esta página hay que seleccionar la opción *Fuel Types* en el menú izquierdo, cuando se abre la pantalla muestra el listado de todos los tipos de combustibles disponibles. Al hacer doble clic sobre cualquiera de los registros se mostrará un formulario con el detalle de los datos y además se permite editar. En el caso de agregar un nuevo tipo de combustible hay que presionar sobre el botón *Add*. Para eliminar un tipo de combustible, presionar sobre el botón en forma de bote de basura junto a cada registro.

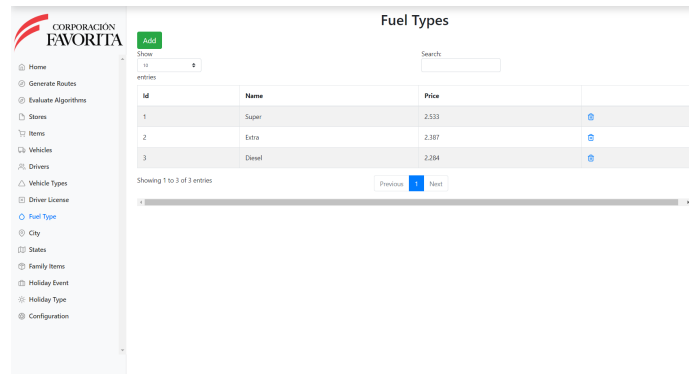


Figura 4.9: Pantalla de tipos de combustible en aplicación web

4.3.9. Ciudades

En esta pantalla se encuentra el mantenimiento de las ciudades que agrupan a las tiendas. La página permite consultar, crear, modificar o eliminar ciudades. Para acceder a esta página hay que seleccionar la opción *Cities* en el menú izquierdo, cuando se abre la pantalla muestra el listado de todas las ciudades disponibles. Al hacer doble clic sobre cualquiera de los registros se mostrará un formulario con el detalle de los datos y además se permite editar. En el caso de agregar una nueva ciudad hay que presionar sobre el botón *Add*. Para eliminar una ciudad, presionar sobre el botón en forma de bote de basura junto a cada registro.

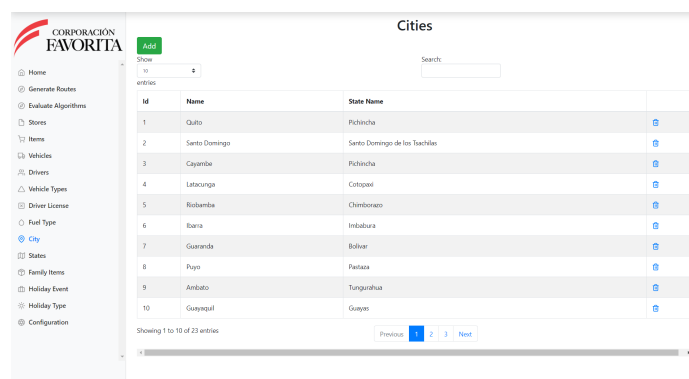
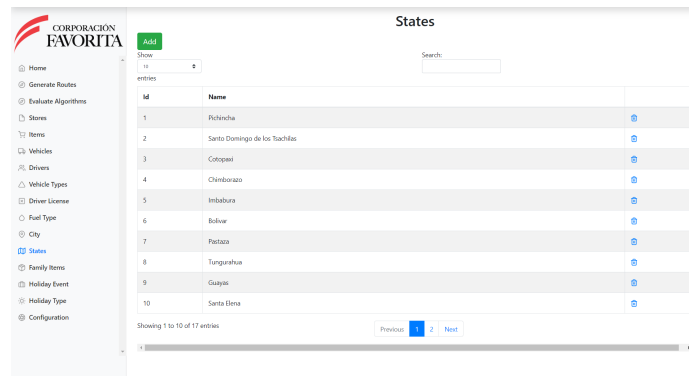


Figura 4.10: Pantalla de ciudades en aplicación web

4.3.10. Provincias

En esta pantalla se encuentra el mantenimiento de las provincias que agrupan las ciudades. La página permite consultar, crear, modificar o eliminar provincias. Para

acceder a esta página hay que seleccionar la opción *States* en el menú izquierdo, cuando se abre la pantalla muestra el listado de todas las provincias disponibles. Al hacer doble clic sobre cualquiera de los registros se mostrará un formulario con el detalle de los datos y además se permite editar. En el caso de agregar una nueva provincia hay que presionar sobre el botón *Add*. Para eliminar una provincia, presionar sobre el botón en forma de bote de basura junto a cada registro.



Id	Name
1	Pichincha
2	Santo Domingo de los Tsáchilas
3	Cotacachi
4	Chimborazo
5	Imbabura
6	Bolivar
7	Pastaza
8	Tungurahua
9	Guayas
10	Santo Elena

Figura 4.11: Pantalla de provincias en aplicación web

4.3.11. Familia de productos

En esta pantalla se encuentra el mantenimiento de las familias de productos en las que se encuentra agrupado cada producto. La página permite consultar, crear, modificar o eliminar familias de productos. Para acceder a esta página hay que seleccionar la opción *Families* en el menú izquierdo, cuando se abre la pantalla muestra el listado de todas las familias de productos disponibles. Al hacer doble clic sobre cualquiera de los registros se mostrará un formulario con el detalle de los datos y además se permite editar. En el caso de agregar una nueva familia de productos hay que presionar sobre el botón *Add*. Para eliminar una familia de productos, presionar sobre el botón en forma de bote de basura junto a cada registro.

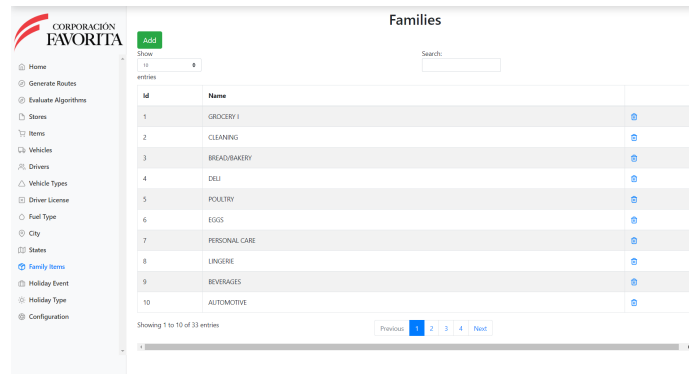


Figura 4.12: Pantalla de familias de productos en aplicación web

4.3.12. Eventos Festivos

En esta pantalla se encuentra el mantenimiento de los eventos festivos que serán usados en la generación de rutas para determinar si la fecha de generación de rutas es festiva. La página permite consultar, crear, modificar o eliminar eventos festivos. Para acceder a esta página hay que seleccionar la opción *Holiday Events* en el menú izquierdo, cuando se abre la pantalla muestra el listado de todos los eventos festivos disponibles. Al hacer doble clic sobre cualquiera de los registros se mostrará un formulario con el detalle de los datos y además se permite editar. En el caso de agregar un nuevo evento festivo hay que presionar sobre el botón *Add*. Para eliminar un evento festivo, presionar sobre el botón en forma de bote de basura junto a cada registro.

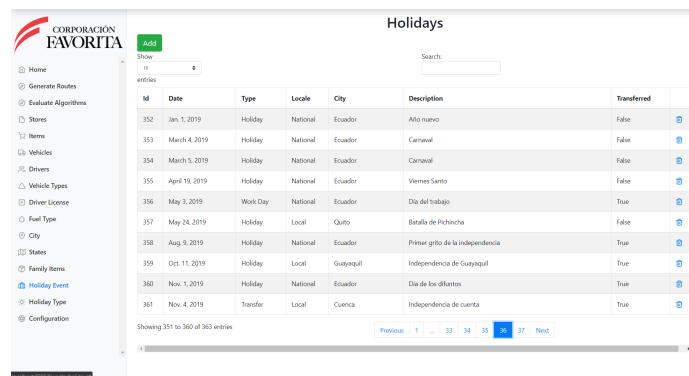


Figura 4.13: Pantalla de días festivos en aplicación web

4.3.13. Configuración

En esta pantalla se encuentra el mantenimiento de la configuración general usada por el algoritmo que permite hacer validaciones y cálculos sobre las funciones objetivo. La página permite modificar y consultar la configuración. Para acceder a esta página debe seleccionar la opción *Configuration* en el menú izquierdo, cuando se despliega el formulario con los parámetros ingresados solamente se permite modificar y consultar. Para guardar los cambios se debe presionar el botón *Submit*.

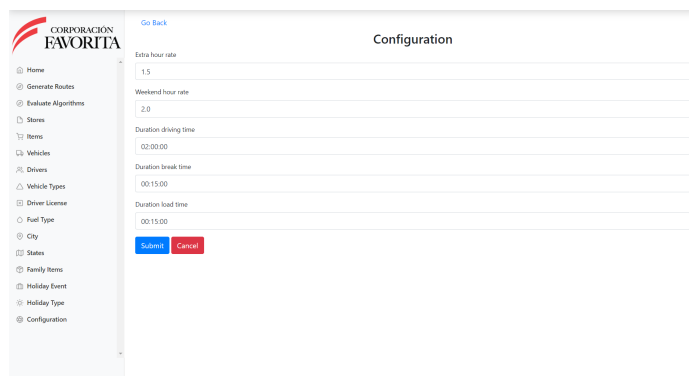


Figura 4.14: Pantalla de configuración general en aplicación web

4.4. Uso del algoritmo en la aplicación web

En la aplicación web existen pantallas que permiten la configuración de los parámetros iniciales del algoritmo genético para después proceder a su ejecución. Estas pantallas hacen que sea más intuitivo para el usuario generar una ruta con una simple selección de pocos parámetros el resto de la configuración necesaria para la búsqueda de la ruta se obtienen internamente. Muchos de los datos requeridos para la ejecución serán consultados desde la base de datos. Los usuarios deben garantizar que todos los datos fueron ingresados previamente a la aplicación, porque de eso depende los resultados obtenidos.

4.4.1. Configuración de la búsqueda de rutas

Esta pantalla permitirá ingresar la configuración inicial antes de iniciar la generación de las rutas usando el algoritmo implementado. Para acceder a la pantalla de generación de rutas hay que presionar la opción *Generate Routes* en el menú de la parte izquierda. Mientras el algoritmo busca la mejor ruta la pantalla permanecerá

bloqueada hasta que el algoritmo finalice y retorne una respuesta. Cuando se encuentre una ruta válida se redirecciona a otra pantalla que muestra los resultados. En caso de error muestra un mensaje detallando el problema

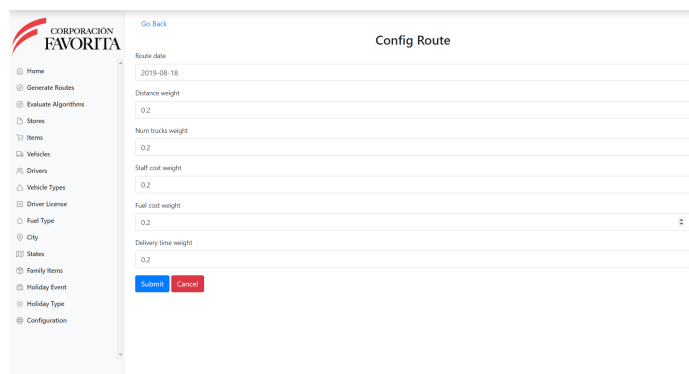


Figura 4.15: Pantalla de configuración para generar rutas en aplicación web

4.4.2. Resultados de la generación de las rutas

En esta pantalla se muestran los resultados obtenidos después de la generación de las rutas. La información se encuentra organizada por rutas, tiendas, conductores, artículos, distancias recorridas y un mapa que muestra la ruta que debe seguir el conductor. Cada vez que se selecciona una ruta, el resto de los datos vinculados se muestran en los diferentes campos.

El primer campo es de selección múltiple, contiene el nombre de la ruta y el vehículo asignado. El segundo campo despliega el nombre de las tiendas y las coordenadas que están asociadas a la ruta. El tercer campo muestra el nombre de los conductores asignados a las rutas seleccionadas. El cuarto campo muestra el listado de productos que se debe entregar, tiene la cantidad y la tienda a la que pertenece. En el quinto campo se muestra la distancia recorrida por cada ruta. Finalmente está implementado un mapa que muestra las rutas seleccionadas en el primer campo, se diferencian por color, también para cada una de las tiendas está numerado según el orden de visita, además se pinta la ruta en carretera que debe seguir el conductor.

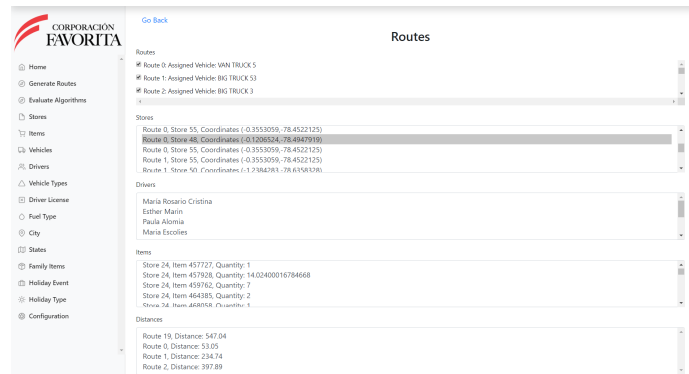


Figura 4.16: Pantalla de rutas generadas en aplicación web

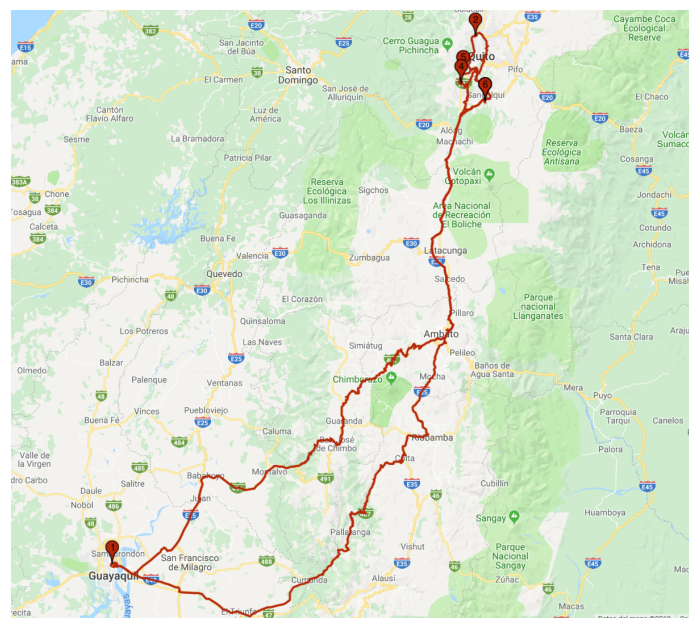


Figura 4.17: Pantalla de rutas generadas usando mapa en aplicación web

4.5. Selección de parámetros en la aplicación web

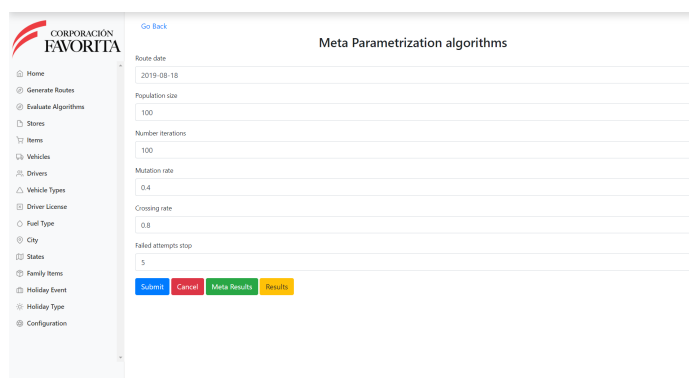
La aplicación permite que se puedan obtener automáticamente configuraciones iniciales para el generador de rutas y encontrar la mejor configuración para cumplir con el objetivo general. Esta pantalla también ejecuta un algoritmo que tiene como objetivo mejorar la calidad de los resultados que se obtienen con el algoritmo genético. Para acceder a la página debe presionar la opción *Evaluate Algorithms* en el menú de

la parte izquierda. Se desplegará un formulario para el ingreso de la configuración de este algoritmo.

La pantalla permanecerá bloqueada hasta que finalice el proceso de evaluación y se retorne un mensaje de error en caso de que falle. Si los resultados son positivos la página se direcciona a una página con los resultados generados por este algoritmo.

En el mismo formulario existen dos botones para poder ver el registro de los resultados generados:

1. **Meta Results** Este botón direcciona hacia una pantalla que muestra el listado de todas las ejecuciones del algoritmo de búsqueda de configuraciones y detalla todos los valores que se utilizaron para la búsqueda.
2. **Results** Este botón direcciona hacia una pantalla que muestra el listado de todas las ejecuciones del algoritmo genético en la generación de las rutas. También, muestra todos los valores utilizados y los resultados obtenidos.



The screenshot shows a web application interface for configuring meta-algorithms. On the left is a sidebar with a navigation menu including 'Home', 'Generate Routes', 'Evaluate Algorithms', 'Stores', 'Items', 'Vehicles', 'Drivers', 'Vehicle Types', 'Driver License', 'Fuel Type', 'City', 'States', 'Family Items', 'Holiday Event', 'Holiday Type', and 'Configuration'. The main content area is titled 'Meta Parametrization algorithms' and contains a 'Go Back' link and several input fields: 'Route date' (2019-08-18), 'Population size' (100), 'Number iterations' (100), 'Mutation rate' (0.4), 'Crossing rate' (0.8), and 'Failed attempts stop' (5). At the bottom of the form are four buttons: 'Submit' (blue), 'Cancel' (red), 'Meta Results' (green), and 'Results' (yellow).

Figura 4.18: Pantalla para ingresar configuración inicial de meta algoritmo en aplicación web

El código de la aplicación se encuentra disponible en el siguiente enlace:
<https://github.com/poncejj/TrabajoFinMasterJJP>

Capítulo 5

Evaluación y análisis de los resultados

Este capítulo detalla la experimentación realizada con el algoritmo genético para poder resolver el problema de distribución de productos en una cadena de supermercados. Están detalladas todas las pruebas realizadas utilizando el algoritmo. Además, está descrito la implementación y evaluación de un meta algoritmo para obtener la mejor configuración inicial en la generación de rutas. También, se dan a conocer los resultados obtenidos en las pruebas realizadas usando las diferentes configuraciones del algoritmo. Al final del capítulo se realiza un análisis comparativo entre los resultados obtenidos al ejecutar el algoritmo genético usando la configuración manual y la mejor configuración obtenida por el meta algoritmo.

5.1. Evaluación algoritmo genético

Las evaluaciones iniciales consisten en ejecutar el algoritmo para la generación de rutas usando una configuración ingresada manualmente, variando los pesos que se otorga a cada objetivo para comprobar que peso tiene más influencia sobre los resultados.

5.1.1. Evaluaciones realizadas

Se realizarán seis evaluaciones diferentes usando la configuración manual. Cada evaluación será distinta en los pesos asignados a los objetivos que se desea minimizar; el tamaño de la población y el número de iteraciones serán constante para todas las evaluaciones; mientras que la tasa de cruce y tasa de mutación son elegidas aleatoriamente.

Cada prueba se diferencia por:

1. Todos los pesos va a ser igual. Se divide uno para los cinco objetivos que se van a evaluar.
2. El peso del objetivo de la distancia es mayor.
3. El peso del objetivo al número de vehículos es mayor.
4. El peso del objetivo al coste de personal es mayor.
5. El peso del objetivo al coste de combustible es mayor.
6. El peso del objetivo al tiempo de entrega es mayor.

Resumen de las evaluaciones realizadas

	Población	Iteraciones	Tasa de cruce	Tasa de mutación	Distancia	Número de vehículos	Costo personal	Costo Combustible	Tiempo de entrega
1	10	2	0.28	0.58	0.20	0.20	0.20	0.20	0.20
2	10	2	0.75	0.14	0.56	0.11	0.11	0.11	0.11
3	10	2	0.53	0.44	0.11	0.56	0.11	0.11	0.11
4	10	2	0.25	0.55	0.11	0.11	0.56	0.11	0.11
5	10	2	1.00	0.75	0.11	0.11	0.11	0.56	0.11
6	10	2	0.12	0.93	0.11	0.11	0.11	0.11	0.56

Tabla 5.1: Tabla que muestra la configuración inicial en cada una de las evaluaciones realizadas con el algoritmo genético en la generación de rutas

5.1.2. Resultados obtenidos

Por cada evaluación se realizó cinco repeticiones porque es una buena práctica llevar a cabo múltiples pruebas antes de aceptar los resultados de un experimento y decidir si se puede confiar o rechazar la configuración probada. Mientras más ensayos son realizados, más confianza se puede tener en los resultados que son similares. Por otro lado, una gran variación de los datos entre los experimentos obliga a la búsqueda de razones de esta disparidad.

Otra razón para realizar varias veces los experimentos es porque los resultados obtenidos son estocásticos y no se pueden replicar en su totalidad debido a la naturaleza del algoritmo genético. Por ejemplo: la población inicial generada en cada ejecución es diferente, la selección de los individuos es aleatorio, la tasa de mutación y cruce en la configuración inicial fueron elegidos al azar dentro de un rango.

Los resultados están presentados en forma de tablas para que sea más fácil su visualización y comprensión. Los valores de la tabla corresponden a los que obtuvieron mejor valor de aptitud en cada evaluación. Primera columna corresponde al número de evaluación asociada. Segunda columna en la tabla presenta el valor de aptitud. Tercera columna la distancia total. Cuarta columna el número total de automóviles usados en la repartición de productos. Quinta columna el costo total de combustible en la ruta. La sexta columna el tiempo máximo utilizado por toda la ruta. Finalmente, el último valor pertenece al tiempo total de ejecución en segundos del algoritmo.

	Valor de aptitud	Distancia Total	Número de Vehículos	Costo de personal	Costo de combustible	Tiempo de reparto	Tiempo de ejecución
1	15130.14	53731.58	147	12216.50	9538.06	17.56	16.36
2	32390.87	53565.12	133	14585.27	8956.23	17.73	17.91
3	8492.05	53093.94	139	13313.91	9307.86	17.74	16.41
4	13562.97	54119.35	138	11679.52	9394.08	17.73	18.15
5	12438.19	53484.34	139	12059.35	9248.65	17.73	16.58
6	8518.06	54593.94	146	12457.05	9377.28	17.66	17.87

Tabla 5.2: Mejores valores de aptitud obtenidos en las evaluaciones del algoritmo genético para la generación de rutas

5.2. Evaluación meta algoritmo

Una de las mejores estrategias que puede ser aplicada para la búsqueda adecuada de parámetros es el uso de meta algoritmos. Un algoritmo externo es usado para buscar los parámetros de configuración de un algoritmo interno. Mejores parámetros suelen ser encontrados comparado con los que se obtienen al configurar manualmente el algoritmo interno. Cada ejecución del meta algoritmo puede evaluar varias veces al algoritmo interno y permite cubrir una mayor área de búsqueda.

Una de las desventajas del uso de meta algoritmos es que el tiempo de ejecución suele ser extenso porque se prueban una gran cantidad de soluciones, pero si se compara con el tiempo empleado en realizar el mismo número de pruebas manualmente, se puede concluir que reduce los tiempos en la selección de nuevas configuraciones porque es un proceso automático y consigue mejores resultados [33].

5.2.1. Diseño del meta algoritmo

El meta algoritmo propuesto en este trabajo va a ser del tipo genético, debido a que su capacidad evolutiva permite encontrar mejores soluciones en el transcurso del tiempo de ejecución. Además, el área del espacio de búsqueda va a ser más extensa y

permite evitar óptimos locales.

El meta algoritmo va a ser ejecutado sobre el algoritmo genético descrito en 2.3. El meta algoritmo también requiere una configuración inicial antes de su ejecución. La adecuada selección de los parámetros iniciales del meta algoritmo influye sobre la calidad de resultados obtenidos al ejecutar el algoritmo interno y ayudará a encontrar una mejor configuración inicial para el algoritmo interno.

Configuración inicial meta algoritmo

La configuración inicial son valores que requiere para su funcionamiento. Los parámetros necesarios son:

- **Fecha de ejecución:** Es la fecha de la que se quiere obtener la ruta óptima.
- **Tamaño de la población:** Número de individuos que va a tener la población inicial del meta algoritmo.
- **Número de iteraciones:** Número de iteraciones que se van a realizar en la etapa evolutiva del meta algoritmo.
- **Tasa de mutación:** Tasa de mutación aplicada en el proceso evolutivo del meta algoritmo.
- **Tasa de cruce:** Tasa de cruce aplicada en el proceso evolutivo del meta algoritmo
- **Número de fallos permitidos:** Es el número de errores que soporta el meta algoritmo antes de que finalice su ejecución. Sirve como condición de parada.

Cromosoma meta algoritmo

El cromosoma usado es un objeto que contiene la configuración inicial para el algoritmo genético. Al momento de generar un individuo los atributos de este objeto son inicializados con valores aleatorios dentro de un rango específico por cada atributo. Los atributos de este objeto son:

- **Fecha de ejecución:** Valor del tipo fecha. Es el mismo que se establece en la configuración inicial del meta algoritmo.
- **Tamaño de la población:** Valor entero con un rango de 10 hasta 100.
- **Número de evoluciones:** Valor entero con un rango de 2 hasta 100.
- **Tasa de mutación:** Valor decimal con un rango de 0 hasta 1.

- **Tasa de cruce:** Valor decimal con un rango de 0 hasta 1.
- **Número de fallos permitidos:** Valor entero con un rango de 1 hasta 10.
- **Peso del objetivo de distancia:** Valor decimal con un rango de 0 hasta 1.
- **Peso del objetivo de número de vehículos:** Valor decimal con un rango de 0 hasta 1.
- **Peso del objetivo de costo de personal:** Valor decimal con un rango de 0 hasta 1.
- **Peso del objetivo de costo de combustible:** Valor decimal con un rango de 0 hasta 1.
- **Peso del objetivo de tiempo de entrega de productos:** Valor decimal con un rango de 0 hasta 1.

Individuo meta algoritmo

El individuo está compuesto por un cromosoma y un valor de aptitud que es obtenido en la función de evaluación.

Individuo 1
Fecha de ejecución: 2019-09-08
Tamaño de la población: 15
Número de evoluciones: 95
Tasa de mutación: 0.91
Tasa de cruce: 0.62
Número de fallos permitidos: 6
Peso del obj. de distancia: 0.22
Peso objetivo núm. de vehículos: 0.03
Peso objetivo costo de personal: 0.27
Peso objetivo costo combustible: 0.26
Peso objetivo tiempo de entrega: 0.21
Valor de aptitud: 15012.43

Figura 5.1: Ejemplo de individuo usado por el meta algoritmo

Población meta algoritmo

La población es un arreglo de individuos. Tiene un tamaño establecido en la configuración inicial. Cada individuo es inicializado aleatoriamente.

Población				
Individuo 1	Individuo 2	Individuo 3	Individuo 4	Individuo 5
Fecha de ejecución: 2019-09-08	Fecha de ejecución: 2019-09-08	Fecha de ejecución: 2019-09-08	Fecha de ejecución: 2019-09-08	Fecha de ejecución: 2019-09-08
Tamaño de la población: 15	Tamaño de la población: 99	Tamaño de la población: 71	Tamaño de la población: 63	Tamaño de la población: 70
Número de evoluciones: 95	Número de evoluciones: 43	Número de evoluciones: 66	Número de evoluciones: 90	Número de evoluciones: 74
Tasa de mutación: 0.91	Tasa de mutación: 0.78	Tasa de mutación: 0.64	Tasa de mutación: 0.55	Tasa de mutación: 0.3
Tasa de cruce: 0.62	Tasa de cruce: 0.99	Tasa de cruce: 0.59	Tasa de cruce: 0.36	Tasa de cruce: 0.39
Número de fallos permitidos: 6	Número de fallos permitidos: 10	Número de fallos permitidos: 1	Número de fallos permitidos: 3	Número de fallos permitidos: 4
Peso del obj. de distancia: 0.22	Peso del obj. de distancia: 0.21	Peso del obj. de distancia: 0.25	Peso del obj. de distancia: 0.35	Peso del obj. de distancia: 0.19
Peso objetivo núm. de vehículos: 0.03	Peso objetivo núm. de vehículos: 0.19	Peso objetivo núm. de vehículos: 0.18	Peso objetivo núm. de vehículos: 0.14	Peso objetivo núm. de vehículos: 0.24
Peso objetivo costo de personal: 0.27	Peso objetivo costo de personal: 0.33	Peso objetivo costo de personal: 0.23	Peso objetivo costo de personal: 0.19	Peso objetivo costo de personal: 0.15
Peso objetivo costo combustible: 0.26	Peso objetivo costo combustible: 0.11	Peso objetivo costo combustible: 0.19	Peso objetivo costo combustible: 0.15	Peso objetivo costo combustible: 0.21
Peso objetivo tiempo de entrega: 0.21	Peso objetivo tiempo de entrega: 0.16	Peso objetivo tiempo de entrega: 0.16	Peso objetivo tiempo de entrega: 0.17	Peso objetivo tiempo de entrega: 0.21
Valor de aptitud: 15012.43	Valor de aptitud: 14750.7	Valor de aptitud: 15015.04	Valor de aptitud: 18460.23	Valor de aptitud: 12720.31

Figura 5.2: Ejemplo de población usado por el meta algoritmo

Evaluación del individuo meta algoritmo

Para evaluar cada individuo del meta algoritmo se debe seguir los siguientes pasos:

1. La población debe estar inicializada para evaluar el algoritmo genético interno.
2. Por cada individuo de la población del meta algoritmo
3. Ejecutar el algoritmo interno usando como configuración inicial el cromosoma asociado al individuo del meta algoritmo que se está evaluando.
4. El valor de aptitud que recibe el individuo del meta algoritmo será el resultado de la ejecución del algoritmo interno.
5. Si la ejecución del algoritmo interno devuelve error, se vuelve a crear un nuevo individuo del meta algoritmo, con nuevos parámetros aleatorios y se contabiliza el error para el manejo de la función de parada.

Selección de los individuos padres meta algoritmo

Para realizar la selección de los individuos padres para usar el operador de cruce se tiene que seguir los siguientes pasos:

1. Evaluar toda la población.
2. Ordenar la población por el valor de aptitud de cada individuo de forma ascendente para que los valores más bajos tengan mayor probabilidad de ser seleccionados. La forma de ordenar es ascendente porque se desea minimizar los objetivos.
3. Se crea un torneo con los diez mejores individuos, para garantizar que la descendencia va a mejorar los resultados obtenidos.

- Elegir los dos individuos padres de forma aleatoria.

Población		Población Ordenada		Torneo	
Nombre Individuo	Valor Aptitud	Nombre Individuo	Valor Aptitud	Nombre Individuo	Valor Aptitud
Individuo 1	17488.47	Individuo 15	5394.49	Individuo 15	5394.49
Individuo 2	15951.38	Individuo 8	5447.23	Individuo 8	5447.23
Individuo 3	16446.73	Individuo 17	5460.69	Individuo 17	5460.69
Individuo 4	13546.41	Individuo 11	5478.6	Individuo 11	5478.6
Individuo 5	13558.38	Individuo 16	5991.09	Individuo 16	5991.09
Individuo 6	11043.43	Individuo 20	6273.19	Individuo 20	6273.19
Individuo 7	12620.11	Individuo 13	7147.67	Individuo 13	7147.67
Individuo 8	5447.23	Individuo 19	7313.86	Individuo 19	7313.86
Individuo 9	11596.52	Individuo 10	8480.55	Individuo 10	8480.55
Individuo 10	8480.55	Individuo 12	9281.18	Individuo 12	9281.18
Individuo 11	5478.6	Individuo 18	10056.29		
Individuo 12	9281.18	Individuo 14	10142.01		
Individuo 13	7147.67	Individuo 6	11043.43		
Individuo 14	10142.01	Individuo 9	11596.52		
Individuo 15	5394.49	Individuo 7	12620.11		
Individuo 16	5991.09	Individuo 4	13546.41		
Individuo 17	5460.69	Individuo 5	13558.38		
Individuo 18	10056.29	Individuo 2	15951.38		
Individuo 19	7313.86	Individuo 3	16446.73		
Individuo 20	6273.19	Individuo 1	17488.47		

Figura 5.3: Ejemplo del operador de selección en el meta algoritmo

Operador de cruce meta algoritmo

El operador de cruce del meta algoritmo está controlado por el valor de *tasa de cruce* que fue ingresado en la configuración inicial del meta algoritmo.

Los siguientes pasos deben realizarse para el operador de cruce

- Los dos individuos padres son elegidos usando el operador de selección.
- La tasa de cruce permite que los dos individuos se crucen.
- Un nuevo individuo es inicializado, pero debe conservar la misma fecha de ejecución.
- Un número binario es usado para decidir los genes que se heredarán al individuo.

Si el número binario es cero el nuevo individuo recibe el valor del del individuo A caso contrario, recibe el valor del individuo B.

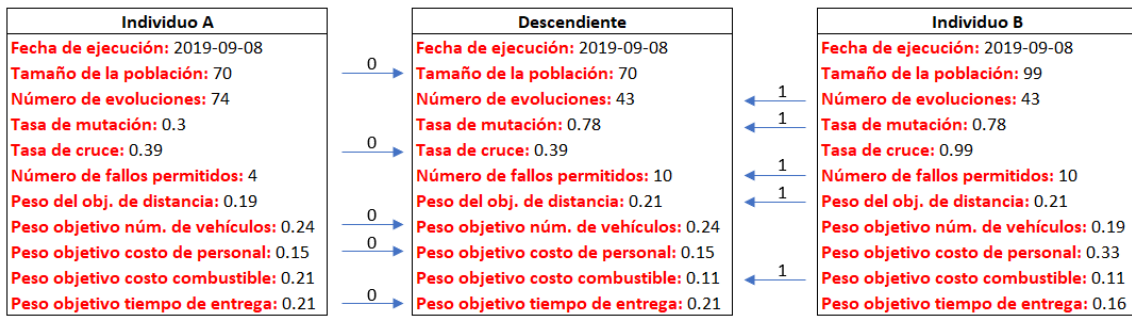


Figura 5.4: Ejemplo del operador de cruce en el meta algoritmo

Operador de mutación meta algoritmo

El operador de mutación del meta algoritmo está controlado por una tasa de mutación establecida en la configuración inicial de los parámetros.

El proceso de mutación se realiza siguiendo los pasos:

1. Validar si la tasa de mutación permite que se realice la mutación.
2. Para cada gen del individuo, un número binario aleatorio decide si se cambia el valor o se conserva.

Si el número es cero, se cambia por un valor aleatorio que esté dentro del rango de ese atributo. Caso contrario, se conserva el valor actual del atributo.

3. Al final se añade el nuevo individuo a la población (aún si es que no se mutó el individuo).

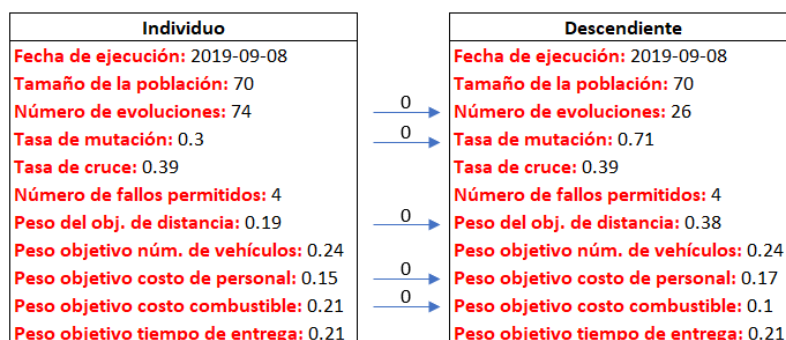


Figura 5.5: Ejemplo del operador de mutación en el meta algoritmo

Reducción de la población meta algoritmo

El proceso de reducción de la población se realiza para conservar siempre el tamaño inicial de la población que fue establecida en la configuración inicial, porque en el proceso evolutivo se añaden nuevos individuos descendientes. Se debe seguir los siguientes para reducir la población.

1. Todos los individuos deben ser evaluados.
2. Ordenar la población por el valor de aptitud de los individuos de forma ascendente.
3. Descartar los individuos que estén fuera del rango de la población inicial.

Población		Población Ordenada		Eliminación de individuos	
Nombre Individuo	Valor Aptitud	Nombre Individuo	Valor Aptitud	Nombre Individuo	Valor Aptitud
Individuo 1	17488.47	Individuo 22	3764.20	Individuo 22	3764.20
Individuo 2	15951.38	Individuo 21	4300.87	Individuo 21	4300.87
Individuo 3	16446.73	Individuo 15	5394.49	Individuo 15	5394.49
Individuo 4	13546.41	Individuo 8	5447.23	Individuo 8	5447.23
Individuo 5	13558.38	Individuo 17	5460.69	Individuo 17	5460.69
Individuo 6	11043.43	Individuo 11	5478.60	Individuo 11	5478.60
Individuo 7	12620.11	Individuo 16	5991.09	Individuo 16	5991.09
Individuo 8	5447.23	Individuo 20	6273.19	Individuo 20	6273.19
Individuo 9	11596.52	Individuo 13	7147.67	Individuo 13	7147.67
Individuo 10	8480.55	Individuo 19	7313.86	Individuo 19	7313.86
Individuo 11	5478.60	Individuo 10	8480.55	Individuo 10	8480.55
Individuo 12	9281.18	Individuo 12	9281.18	Individuo 12	9281.18
Individuo 13	7147.67	Individuo 18	10056.29	Individuo 18	10056.29
Individuo 14	10142.01	Individuo 14	10142.01	Individuo 14	10142.01
Individuo 15	5394.49	Individuo 6	11043.43	Individuo 6	11043.43
Individuo 16	5991.09	Individuo 9	11596.52		
Individuo 17	5460.69	Individuo 7	12620.11		
Individuo 18	10056.29	Individuo 4	13546.41		
Individuo 19	7313.86	Individuo 5	13558.38		
Individuo 20	6273.19	Individuo 2	15951.38		
Individuo 21	4300.87	Individuo 3	16446.73		
Individuo 22	3764.20	Individuo 1	17488.47		

Tamaño de población: 15

Figura 5.6: Ejemplo de eliminación de individuos en el meta algoritmo

Condición de parada meta algoritmo

El algoritmo finaliza cuando:

1. Todo el proceso de inicialización de la población se ha generado correctamente, se ha cumplido el número de iteraciones de evolución entre los cromosomas y se ha encontrado una solución válida para el problema. La solución que devuelve el algoritmo es aquella que tenga el menor valor de aptitud porque el problema de optimización desea minimizar los objetivos.
2. Durante el proceso de inicialización de la población se usa un contador para determinar el número de veces que un individuo no válido fue generado. Si el

valor del contador es igual al parámetro *Número de errores permitidos* se retorna un mensaje de error indicando que no existe solución para el problema.

3. Un contador es usado para controlar los errores obtenidos durante el proceso evolutivo al realizar mutación o cruce. Si el contador es igual a la variable de *Número de errores permitidos* se retorna un mensaje de error indicando que no existe solución para el problema.

5.2.2. Evaluaciones realizadas sobre meta algoritmo

Las evaluaciones del meta algoritmo se realizan cambiando su configuración inicial. El tamaño de la población y un número de iteraciones son muy pequeños y en cada evaluación realizada aumentarán, se realizan varias repeticiones de cada evaluación y en cada repetición la tasa de cruce y mutación aumentan hasta llegar al máximo (uno). Los parámetros adicionales serán constantes durante todas las pruebas realizadas. Varias evaluaciones con múltiples repeticiones son realizadas para poder explorar una mayor área en el espacio de búsqueda.

Los valores usados en cada evaluación son los siguientes:

Primera evaluación

	Población	Iteraciones	Tasa de cruce	Tasa de mutación	Número de fallos máx.
1	2	2	0.20	0.20	5
2	2	2	0.40	0.40	5
3	2	2	0.60	0.60	5
4	2	2	0.80	0.80	5
5	2	2	1.00	1.00	5

Tabla 5.3: Parámetros usados en la primera evaluación del meta algoritmo

Segunda evaluación

	Población	Iteraciones	Tasa de cruce	Tasa de mutación	Número de fallos máx.
6	5	5	0.20	0.20	5
7	5	5	0.40	0.40	5
8	5	5	0.60	0.60	5
9	5	5	0.80	0.80	5
10	5	5	1.00	1.00	5

Tabla 5.4: Parámetros usados en la segunda evaluación del meta algoritmo

Tercera evaluación

	Población	Iteraciones	Tasa de cruce	Tasa de mutación	Número de fallos máx.
11	10	10	0.20	0.20	5
12	10	10	0.40	0.40	5
13	10	10	0.60	0.60	5
14	10	10	0.80	0.80	5
15	10	10	1.00	1.00	5

Tabla 5.5: Parámetros usados en la tercera evaluación del meta algoritmo

Cuarta evaluación

	Población	Iteraciones	Tasa de cruce	Tasa de mutación	Número de fallos máx.
16	20	20	0.20	0.20	5
17	20	20	0.40	0.40	5
18	20	20	0.60	0.60	5
19	20	20	0.80	0.80	5
20	20	20	1.00	1.00	5

Tabla 5.6: Parámetros usados en la cuarta evaluación del meta algoritmo

5.2.3. Resultados obtenidos

Los resultados presentados a continuación pertenecen a las evaluaciones realizadas utilizando el meta algoritmo. Pertenecen a los mejores valores de aptitud obtenidos por cada evaluación.

Las columnas de la tabla representan:

1. Número de evaluación a la que pertenece el resultado.
2. El mejor valor de aptitud obtenido en la ejecución del meta algoritmo.
3. Tiempo total de ejecución del algoritmo medido en segundos.

	Mejor valor de aptitud	Tiempo de Ejecución
1	13546.41	546.89
2	5447.23	89990.23
3	5394.49	3987.02
4	5460.68	8243.80

Tabla 5.7: Mejores resultados obtenidos durante las evaluaciones realizadas con el meta algoritmo para la obtención de la configuración inicial

5.3. Evaluación algoritmo genético usando la mejor configuración

Esta evaluación consiste en usar la mejor configuración que fue encontrada por el meta algoritmo sobre el algoritmo genético. Varias repeticiones de la evaluación fueron realizadas porque los algoritmos genéticos tienen resultados estocásticos.

Los siguientes parámetros corresponden a la mejor configuración:

- *Tamaño de la población:* 27
- *Número de iteraciones:* 51
- *Tasa de cruce:* 0.64
- *Tasa de mutación:* 0.65
- *Peso para el objetivo distancia total:* 0.07
- *Peso para el objetivo número de vehículos:* 0.25
- *Peso para el objetivo costo de personal:* 0.40
- *Peso para el objetivo costo de combustible:* 0.05
- *Peso para el objetivo tiempo de entrega:* 0.22

5.3.1. Resultados obtenidos

En la tabla se encuentran representados los resultados obtenidos en la ejecución del algoritmo genético usando la configuración descrita en el párrafo anterior.

Las columnas de la tabla representan:

1. Número de repetición de la evaluación
2. Valor de aptitud obtenido después de la ejecución del algoritmo.
3. Distancia total recorrida en todas las rutas.
4. Número de vehículos usados para el reparto de los productos.
5. Costo total de combustible de todas las rutas.

6. Costo total de personal para poder operar los vehículos.
7. Tiempo que se demora en realizarse el reparto de los productos.
8. Tiempo total de la ejecución del algoritmo medido en segundos.

	Valor de aptitud	Distancia Total	Número de Vehículos	Costo de personal	Costo de combustible	Tiempo de reparto	Tiempo de ejecución
1	6552.02	48889.73	148	6491.84	8533.02	17.42	246.68
2	6829.16	47613.32	137	7456.84	8140.88	17.73	246.38
3	6729.64	44187.43	143	7865.47	7667.84	17.68	243.60
4	6546.93	48869.91	144	6507.26	8356.35	17.48	248.99
5	6783.01	50042.59	145	6831.80	8788.51	17.30	249.93

Tabla 5.8: Resultados obtenidos usando la mejor configuración inicial encontrada por el meta algoritmo

5.4. Análisis de los resultados

- Un mejor valor de aptitud se obtuvo cuando los pesos de los objetivos de tiempo de reparto y número de vehículos fueron mejor ponderados. Por ejemplo, si se compara las evaluaciones realizadas con la configuración manual los valores de aptitud son menores en la evaluación 3 y 6 que son donde estos pesos tenían mayor valor. Este resultado se debe a que los dos objetivos son los que obtienen los resultados más bajos en la evaluación individual y si se les asigna un mayor peso causa que el valor de aptitud se reduzca significativamente porque los otros objetivos pierden importancia.
- El resultado en la evaluación de la distancia total en las pruebas usando la configuración manual para el algoritmo genético es mayor comparada con los resultados obtenidos en las evaluaciones usando la mejor configuración. En el primer caso el rango de la distancia está entre 51000 km y 56000 km. En cambio, con las últimas pruebas el rango se redujo a 44000 km hasta 50000 km.
- El resultado del número de vehículos en todas las evaluaciones es muy parecido, el mínimo valor obtenido es 128 y el máximo es de 150 vehículos. Los mejores resultados para este objetivo se encuentran en las evaluaciones con configuración manual.
- El resultado del objetivo costo de personal en las ejecuciones de configuración manual varía entre los \$10.500 y los \$14.500. Mientras que, en las últimas pruebas usando la mejor configuración se obtiene un rango desde \$6.500 hasta \$7.500.

También se puede observar que la mejor configuración asigna un peso mayor a este objetivo.

- Los resultados del costo de combustible en las primeras evaluaciones realizadas con configuración manual tienen un precio muy similar, también, se observa que el rango de los valores fluctúa entre los \$8.900 y \$9.500. En las últimas ejecuciones se puede ver que hubo una pequeña mejora de los valores obtenidos, debido a que el rango está entre \$7.699 y \$8.800.
- El tiempo de reparto en todas las evaluaciones realizadas no varía mucho. Si la configuración cambia el tiempo de ejecución de la ruta va a ser similar. Esto se debe a que este objetivo se evalúa con el mayor tiempo de reparto entre todas las rutas, porque se entiende que contiene a los demás y además la ruta que más tiempo tarda atiende a las tiendas más lejanas.
- Los tiempos de ejecución de las evaluaciones es muy similar. Pero se observa que cuando existe un mayor tamaño de la población este valor se incrementa. Por ejemplo, en la última evaluación el tiempo de ejecución es mucho mayor con respecto a las primeras evaluaciones porque la mejor configuración tiene una población mayor. Además, si se incrementa el número de iteraciones el tiempo también aumentará ya que es directamente proporcional.
- El meta algoritmo obtuvo mejores resultados en la evaluación del algoritmo interno cuando usó una población más grande porque busca sobre un mayor espacio y puede localizar mejores soluciones.
- El uso del meta algoritmos produce un mayor coste computacional y por lo tanto los tiempos de evaluación también aumentan. Esto sucede en especial cuando el tamaño de la población aumenta del meta algoritmo aumenta. Por ejemplo, si comparamos la primera evaluación del meta algoritmo donde la población es muy reducida se obtuvo un tiempo máximo de 1780 segundos, comparado con la última ejecución donde la población del meta algoritmo fue la máxima y el tiempo que tardó fue 10700 segundos.

Capítulo 6

Conclusiones y Recomendaciones

6.1. Conclusiones

En esta sección se plasman las conclusiones obtenidas al finalizar el desarrollo de este trabajo. Las conclusiones fueron generadas a partir de los resultados obtenidos en la evaluación y de los puntos más destacados en todo el proceso realizado.

1. En esta investigación se obtuvieron mejores resultados cuando se utilizó la configuración inicial generada por el meta algoritmo comparado con los resultados con la configuración manual del algoritmo porque en las evaluaciones realizadas se obtiene un mejor valor de aptitud con la mejor configuración generada por el meta algoritmo. Además, se minimizaron la mayoría de los resultados individuales de los objetivos. También, se determinó que los resultados mejoran si el tamaño de la población usado en el meta algoritmo aumenta porque si existen más individuos se prueba más veces el algoritmo genético y aumenta el espacio de búsqueda, pero el tiempo de cómputo aumenta considerablemente porque se debe procesar más ejecuciones del algoritmo interno.
2. Existen muchos objetivos que son contradictorios como por ejemplo *número de vehículos* y la *distancia recorrida* porque si se intenta reducir el número de vehículos aumenta la distancia para poder cumplir con las entregas. También, el objetivo *tiempo de reparto* es contradictorio con el objetivo de *número de vehículos* porque si se disminuye la cantidad de automotores en el reparto los tiempos aumentan.
3. Existe mucha dependencia entre los trabajadores y los camiones porque, si no hay la cantidad de conductores para operar los camiones estos no son tomados en cuenta en las rutas por las validaciones realizadas y en muchos casos no se puede satisfacer el problema porque quedan tiendas pendientes. Para cumplir

con las rutas siempre debe existir mayor número de conductores que los vehículos teniendo en consideración los permisos de conducir asociados a cada conductor.

4. La aplicación web es perfectamente adaptable a cualquier empresa que tenga problemas de optimización en logística y cadenas de suministro, porque los datos de entrada son genéricos y se pueden configurar para trabajar con diversos tipos de negocios.
5. La aplicación está pensada para facilitar el trabajo realizado por el personal encargado de la coordinación logística porque automatiza el proceso de generación de rutas, permite ahorrar tiempo y obtener resultados más precisos.
6. Los recursos podrán ser mejor aprovechados, porque el espacio desperdiciado en los camiones se reducirá, ya que el algoritmo intenta ocupar el máximo posible; los conductores reducirán los tiempos de entrega porque seguirán el camino trazado por la aplicación; se reducirá el número de camiones asignados a la ruta y reducirá los costos que genera el reparto de productos.
7. Con el desarrollo de este trabajo se evidencia que es posible implementar algoritmos de inteligencia artificial para brindar soluciones a los problemas de optimización en logística y cadenas de suministro, debido a que se encontraron muchas soluciones factibles y con resultados bastante optimizados.

6.2. Recomendaciones y trabajo futuro

En el trabajo realizado se diseñó un algoritmo que obtiene muy buenos resultados, pero aún existe la posibilidad de mejorarlos y con respecto a la aplicación web se trató de abarcar la mayor cantidad de funcionalidades que se pueden implementar, pero existe aún trabajo que se puede desarrollar para mejorar su funcionamiento.

- Para obtener una mejor configuración inicial se podría realizar nuevos experimentos. Se propone aumentar el tamaño de la población del meta algoritmo para comprobar si se logra mejorar aún más los resultados obtenidos hasta el momento.
- Agregar más restricciones al problema, por ejemplo: la capacidad de almacenamiento de la bodega central y de las tiendas, los horarios en los que se pueda cargar y descargar los productos, tener un registro de los conductores que ya cumplieron con su jornada laboral para dar prioridad a otros que estén disponibles.

- Se propone implementar en la aplicación un algoritmo para la predicción de las ventas futuras para determinar la demanda de las tiendas de una manera más precisa.
- Se puede añadir a los vehículos GPS para conectarlo con la aplicación y poder ver en tiempo real donde se encuentra cada automotor.
- Se recomienda mantener los datos de la aplicación actualizados debido a que de estos depende el éxito de la generación de rutas.

Bibliografía

- [1] Zijm, Henk, Matthias Klumpp, Sunderesh Heragu y Alberto Regattieri: *Operations, Logistics and Supply Chain Management: Definitions and Objectives*. En *Operations, Logistics and Supply Chain Management*, páginas 27–42. Springer, 2019.
- [2] Vélez, Mario César y José Alejandro Montoya: *Metaheurísticos: una alternativa para la solución de problemas combinatorios en administración de operaciones*. Revista Eia, 8:99–115, 2007.
- [3] Bramel, Julien y David Simchi-Levi: *The logic of logistics: theory, algorithms, and applications for logistics management*. Springer, 1997.
- [4] Slimani, Ilham, Ilhame El Farissi y Said Achchab: *Configuration and implementation of a daily artificial neural network-based forecasting system using real supermarket data*. International Journal of Logistics Systems and Management, 28(2):144–163, 2017.
- [5] Ulrich, Keith: *DHL Open Innovation: Program for the Development, Deployment and Promotion of Innovative Solutions in Logistics*. En *Strategies and Communications for Innovations*, páginas 305–317. Springer, 2011.
- [6] Favorita, Corporación la: *Año 2018*. <http://www.corporacionfavorita.com/acerca-de/resumen-2018/>, visitado el 2019-09-01.
- [7] Nocedal, Jorge y Stephen Wright: *Numerical optimization*. Springer Science & Business Media, 2006.
- [8] Dantzig, George B y John H Ramser: *The truck dispatching problem*. Management science, 6(1):80–91, 1959.
- [9] Deb, Kalyanmoy: *Multi-objective optimization*. En *Search methodologies*, páginas 403–449. Springer, 2014.

- [10] Miettinen, Kaisa y P Preface By-Neittaanmaki: *Evolutionary algorithms in engineering and computer science: recent advances in genetic algorithms, evolution strategies, evolutionary programming, GE*. John Wiley & Sons, Inc., 1999.
- [11] Kramer, Oliver: *Genetic algorithm essentials*, volumen 679. Springer, 2017.
- [12] John, Holland: *Holland, Adaptation in natural and artificial systems*, 1992.
- [13] Goldberg, David E y John H Holland: *Genetic algorithms and machine learning*. Machine learning, 3(2):95–99, 1988.
- [14] Davis, Lawrence: *Handbook of genetic algorithms*. Van Nostrand Reinhold, 1991.
- [15] Michalewicz, Zbigniew y Cezary Z Janikow: *Handling constraints in genetic algorithms*. En *ICGA*, páginas 151–157, 1991.
- [16] Reeves, Colin R: *A genetic algorithm for flowshop sequencing*. Computers & operations research, 22(1):5–13, 1995.
- [17] Mirjalili, Seyedali: *Genetic algorithm*. En *Evolutionary Algorithms and Neural Networks*, páginas 43–55. Springer, 2019.
- [18] Syswerda, Gilbert: *Simulated crossover in genetic algorithms*. En *Foundations of genetic algorithms*, volumen 2, páginas 239–255. Elsevier, 1993.
- [19] Karafotias, Giorgos, Mark Hoogendoorn y Ágoston E Eiben: *Parameter control in evolutionary algorithms: Trends and challenges*. IEEE Transactions on Evolutionary Computation, 19(2):167–187, 2014.
- [20] De Landgraaf, WA, AE Eiben y Volker Nannen: *Parameter calibration using meta-algorithms*. En *2007 IEEE Congress on Evolutionary Computation*, páginas 71–78. IEEE, 2007.
- [21] Baker, Barrie M y MA Ayechev: *A genetic algorithm for the vehicle routing problem*. Computers & Operations Research, 30(5):787–800, 2003.
- [22] Ho, William, George TS Ho, Ping Ji y Henry CW Lau: *A hybrid genetic algorithm for the multi-depot vehicle routing problem*. Engineering Applications of Artificial Intelligence, 21(4):548–557, 2008.
- [23] Gendreau, Michel, Gilbert Laporte y Jean Yves Potvin: *Metaheuristics for the capacitated VRP*. En *The vehicle routing problem*, páginas 129–154. SIAM, 2002.
- [24] Robusto, C Carl: *The cosine-haversine formula*. The American Mathematical Monthly, 64(1):38–40, 1957.

- [25] Favorita, Corporación la: *Corporación Favorita Grocery Sales Forecasting*. <https://www.kaggle.com/c/favorita-grocery-sales-forecasting>, visitado el 2019-09-01.
- [26] Turismo Ecuador, Ministerio de: *Calendario de Feriados Ecuador 2019*. <https://www.turismo.gob.ec/wp-content/uploads/2019/01/FERIADOS-2019.pdf>, visitado el 2019-09-01.
- [27] Favorita, Corporación la: *Locales Supermaxi*. <http://www.supermaxi.com/locales/>, visitado el 2019-09-01.
- [28] nacional, EP Petroecuador gerencia de comercialización: *Precios de venta a nivel de terminal para las comercializadoras calificadas y autorizadas a nivel nacional*. <https://www.eppetroecuador.ec/wp-content/uploads/downloads/2019/06/ESTRUCTURA-DE-PRECIOS-JUNIO-2019-MENSUAL-SNI-DEL-06-AL-12-DE-JUNIO-2019.pdf>, visitado el 2019-09-01.
- [29] Asamblea Constituyente, Pleno de la: *Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial*. Decreto Ejecutivo 1196, 2008.
- [30] Corporation, Oracle: *MySQL 8.0 Reference Manual*, 2019. <https://downloads.mysql.com/docs/refman-8.0-en.a4.pdf>, visitado el 2019-09-01.
- [31] Python: *BeginnersGuide*, 2019. <https://wiki.python.org/moin/BeginnersGuide>, visitado el 2019-09-01.
- [32] Foundation, Django Software: *Django Documentation*, 2019. <https://buildmedia.readthedocs.org/media/pdf/django/2.2.x/django.pdf>, visitado el 2019-09-01.
- [33] Labadie, Nacima, Christian Prins, Caroline Prodhon, Nicolas Monmarché y Patrick Siarry: *Metaheuristics for Vehicle Routing Problems*. Wiley Online Library, 2016.