DESARROLLO DE UN MODELO HIBRIDO ENTRE LA METAHEURISTICA “COLONIA DE HORMIGAS” Y EL ALGORITMO GENETICO MODIFICADO DE CHU-BEASLEY APLICADO A LA RECOLECCION DE DESECHOS EN LOS HOSPITALES DE PEREIRA

GUSTAVO ADOLFO GARCIA LONDOÑO

ANDRES FELIPE RAMIREZ VELEZ

UNIVERSIDAD TECNOLOGICA DE PEREIRA

PROGRAMA DE INGENIERIA DE SISTEMAS Y COMPUTACION

PEREIRA

2019

DESARROLLO DE UN MODELO HIBRIDO ENTRE LA METAHEURISTICA "COLONIA DE HORMIGAS" Y EL ALGORITMO GENETICO MODIFICADO DE CHU-BEASLEY APLICADO A LA RECOLECCION DE DESECHOS EN LOS HOSPITALES DE PEREIRA

GUSTAVO ADOLFO GARCIA LONDOÑO

ANDRES FELIPE RAMIREZ VELEZ

TRABAJO DE GRADO PARA OPTAR POR EL TITULO DE INGENIERO DE SISTEMAS Y COMPUTACION

DIRECTOR

Ph.D GUILLERMO ROBERTO SOLARTE MARTINEZ

UNIVERSIDAD TECNOLOGICA DE PEREIRA

PROGRAMA DE INGENIERIA DE SISTEMAS Y COMPUTACION

PEREIRA

2019

Contenido

AGRADECIMIENTOS

Queremos darle gracias a nuestras familias por el apoyo incondicional durante toda nuestra formación como profesionales, a Dios y a la Virgen por guiar nuestros caminos y siempre darnos la protección espiritual y mental, a la Universidad Tecnológica de Pereira quien nos brindó la oportunidad de crecer profesionalmente, a nuestros compañeros quienes estuvieron con nosotros desde el principio de nuestros estudios, al PhD Guillermo Roberto Solarte Martínez por su ayuda durante el desarrollo de este proyecto y permitirnos generar nuevos conocimientos en el campo de la inteligencia artificial, a la Ms Claudia Patricia Arias Hernández por su gran apoyo y conocimiento que permitieron una excelente orientación durante el desarrollo de este trabajo.

Muchas gracias.

RESUMEN

El problema de ruteo de vehículos (VRP) ha sido uno de los problemas de gran importancia e influencia a lo largo de la historia, por tal razón las investigaciones y estudios se han enfocado en implementar diversos algoritmos que permitan encontrar una solución óptima a estos problemas, El VRP es una versión extendida de uno de los problemas más ampliamente estudiados como lo es el problema del agente viajero (TSP), el objetivo que se persigue en este problema es minimizar la distancia total recorrida, cuando uno o varios agentes deben visitar un numero de nodos y regresar al punto inicial de partida; debido a que estos problemas son considerados difíciles de resolver y dentro de la optimización combinatoria son conocidos como problemas NP- Hard, pues de estos no se obtiene una solución de manera eficiente; así mismo dentro de la teoría de la complejidad computacional pertenecen a la clase NP-Completos, lo que indica que no se puede garantizar encontrar la mejor solución en un tiempo de computo razonable, ya que este aumenta exponencialmente, generando así la búsqueda de soluciones aproximadas.

En este trabajo se llevara a cabo la implementación del algoritmo genético de Chu-Beasley (AGCB) y la metaheurística de Optimización por Colonia de Hormigas (ACO) de manera que se desarrolle un modelo hibrido entre estos dos métodos, proporcionando una solución al problema de VRP aplicado a la recolección de desechos de los hospitales de la ciudad de Pereira teniendo en cuenta la restricción del tiempo, convirtiéndose en un VRPTW (Vehicle Routing

Problem With Time Windows) con el objetivo de minimizar la distancia total recorrido de la flota encargada de las rutas de recolección.

ASBTRACT

The problem of vehicle routing (VRP) has been one of the problems of great importance and influence throughout history, for this reason research and studies have focused on implementing various algorithms that allow finding an optimal solution to these problems The VRP is an extended version of one of the most widely studied problems such as the problem of the traveling agent (TSP), the objective pursued in this problem is to minimize the total distance traveled, when one or more agents must visit a number of nodes and return to the starting point of departure; because these problems are considered difficult to solve and within the combinatorial optimization they are known as NP-Hard problems, since a solution cannot be obtained efficiently; Likewise, within the theory of computational complexity they belong to the NP-Complete class, which indicates that the best solution cannot be guaranteed in a reasonable computation time, since this increases exponentially, thus generating the search for approximate solutions .

In this work, the implementation of the Chu-Beasley genetic algorithm (AGCB) and the Metaheuristic of Ant-Colony Optimization (ACO) will be carried out so that a hybrid model is developed between these two methods, providing a solution to the problem of VRP applied to the collection of waste from the hospitals of the city of Pereira taking into account the restriction of time, becoming a VRPTW

(Vehicle Routing Problem With Time Windows) in order to minimize the total distance traveled by the fleet in charge of collection routes

INTRODUCCION

Las entidades hospitalarias generan residuos que deben ser tratados de manera especial, de acuerdo a su clasificación, la cual comprende residuos *No Peligrosos* y *Peligrosos*, en donde los primeros no presentan un riesgo para la salud humana y/o el medio ambiente y dentro de los cuales se encuentran aquellos que son biodegradables, reciclables, inertes, ordinarios o comunes, y los *Peligrosos* que se caracterizan por ser infecciosos, combustibles, inflamables, explosivos, reactivos, radiactivos, volátiles, corrosivos y/o tóxicos; con el fin de darle un manejo integral a estos materiales se crea el Manual de Procedimientos para la Gestión Integral de Residuos bajo la Resolución número 01164 de 2002[1]. Dentro de dicha resolución se hace referencia a la recolección, en donde esta actividad se debe desarrollar en el tiempo oportuno, por parte de empresas de recolección, transporte, aprovechamiento, tratamiento y disposición final de los mismos, mediante la utilización de la tecnología apropiada, a la frecuencia requerida y con el monitoreo correspondiente a los procedimientos establecidos por los Ministerios del Medio Ambiente y de la Protección Social.

Con el fin de garantizar que el proceso de recolección se lleve a cabo de manera eficiente y oportuna, es necesario que las empresas cuenten con una programación de rutas adecuada, que permita no solo visitar a todos los centros

[1] Consultado de http://biblioteca.saludcapital.gov.co/img_upload/03d591f205ab80e521292987c313699c/resolucion-1164-de-2002.pdf

hospitalarios, sino que también lo realice en los tiempos establecidos, con el fin de evitar la proliferación de bacterias que puedan contaminar el ambiente y causar enfermedades, por lo tanto, estas empresas deben contar con una programación de rutas que tenga en cuenta las ventanas de tiempo, la capacidad de los vehículos y adicional a esto generar recorridos a través de rutas cortas que garanticen el servicio en todos los centros de salud.

En términos computacionales, toda esta información se convierte en grandes volúmenes de datos que se deben almacenar para la toma de decisiones, teniendo en cuenta la distancia a recorrer, el número de vehículos, la capacidad y factores ambientales. Todo esto conlleva a un aumento computacional en costos y complejidad algorítmica (Balari, 2005).

El VRP es un problema TSP, pero en este los agentes son reemplazados por vehículos, se incluyen las demandas de cada nodo y se especifica la capacidad de carga de cada vehículo individual (Mohapatra, 2014). Los problemas de este tipo son estudiados de acuerdo con su complejidad, los recursos comúnmente estudiados en complejidad computacional son: el tiempo (aproximación al número de pasos de ejecución que un algoritmo emplea para resolver un problema), y el espacio (aproximación a la cantidad de memoria utilizada para resolver el problema). De acuerdo con su complejidad los problemas se clasifican en L (lineal), NL (no lineal), P (polinomial), P-Complete (polinomial completo), NP (no polinomial), NP-Complete (No polinomial completo), NP-Hard (no polinomial duro). Debido a la dimensión y complejidad que presenta el VRP, este

es considerado un tipo de problema NP-Hard esto es, no existe un algoritmo que en tiempo polinomial pueda resolver cualquier instancia del problema.

Muchos investigadores a lo largo de los años han implementado distintas soluciones para tratar de dar soluciones óptimas a este tipo de problemas, dentro de los cuales se encuentran los algoritmos o modelos híbridos que dan la posibilidad de encontrar soluciones más eficientes.

En este trabajo se desarrolla la implementación del algoritmo genético de Chu-Beasley (AGCB), el cual presenta un proceso evolutivo altamente eficiente, que siendo un algoritmo genético básico posee la característica fundamental de mantener durante todo el proceso la diversidad de los individuos presentes en la población, reemplazando en cada generación un solo individuo, siempre que este cumpla con las condiciones establecidas de optimalidad y factibilidad. Así como también se desarrolla la metaheurística de Optimización por colonia de hormigas (ACO) que fue presentada por Marco Dorigo en su tesis doctoral en 1992 donde básicamente, se trata de una técnica probabilística para resolver problemas computacionales complejos que pueden ser reducidos a la búsqueda de buenos caminos en grafos para solucionar problemas de optimización combinatoria; de manera que, la unión de ambos algoritmos pueda proporcionar una alternativa para encontrar la mejor ruta de recolección de los residuos hospitalarios minimizando la distancia total recorrida y cumpliendo con la restricción del tiempo.

1. ESTADO DEL ARTE.

1.1. Análisis e implementación del algoritmo genético de Chu-Beasley para resolver el problema del agente viajero (TSP) y su variante, el problema de Rutas de Vehículo (VRP)

Autores: Claudia Patricia Arias

Descripción. En este trabajo se lleva a cabo la implementación del algoritmo genético de Chu-Beasley (AGCB), El cual presenta un proceso altamente eficiente, que se desarrolla de forma clásica llevando a cabo los procesos de evolución natural como la selección, en el cual se determinan los padres aptos para pasar a la siguiente generación; la recombinación, en donde los cromosomas de los padres se comparten para generar nuevos individuos y así determinar cuál de los nuevos individuos sigue en el proceso, y finalmente la mutación en donde se controla la factibilidad nuevamente y se emplea una tasa que es ajustable dentro de los parámetros generales del algoritmo.

Aportes a la tesis: El artículo es de gran ayuda en la comprensión de los conceptos esenciales dentro de los algoritmos genéticos y conocer el proceso que usan como aporte a los problemas del TSP y de VRP, así como el algoritmo genético modificado de Chu-Beasley; el cual ha sido empleado en la solución de problemas de gran complejidad como son los pertenecientes a la clase NP-Hard, debido a su capacidad para dar soluciones óptimas.

1.2. Solución del problema de ruteo de vehículos dependientes del tiempo utilizando un algoritmo genético modificado.

Autores: Fredy Alexander Guasmayan Guasmayan.

Descripción: El autor propone resolver el problema de ruteo de vehículos con dependencia del tiempo (TDCVRP) como una variante del ruteo de vehículos con capacidad fija, el cual considera el tiempo u horario en el que se hace el recorrido de la ruta de los vehículos para hacer la entrega de productos a cada cliente, la restricción adicional del tiempo es de gran importancia en los problema de ruteo reales, puesto que los costos en el recorrido de la o las rutas de los vehículos son elevado debido al tráfico y la congestión en determinadas horas del día, El objetivo es implementar una metodología basada en el algoritmo genético modificado de Chu-Beasley aplicado a la empresa de distribución de productos lácteos COLÁCTEOS en la ciudad de Pasto.

Aportes a la tesis: El artículo presenta una solución óptima usando el algoritmo genético de Chu-Beasley empleando una flota homogénea y una dependencia en los horarios de entrega, se tiene en cuenta otros factores como la hora pico, lo cual es de gran importancia para comprender el concepto de ventanas de tiempo y los aportes que se dan en el desarrollo del modelo matemático para el problema que se quiere dar solución.

1.3. A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows.

Autores: Luca Maria Gambardella, Éric Taillard y Giovanni Agazzi.

Descripción: En el artículo los autores resuelven el problema de enrutamiento con ventanas de tiempo empleando la semejanza del comportamiento de las hormigas usando dos hormigueros artificiales que colaboraban jerárquicamente mediante el intercambio de información en la actualización de las feromonas.

Aportes a la tesis: El artículo presenta gran relevancia para poder entender los inicios de esta la metaheurística Colonia de hormiga, que emplea la optimización de rutas con ventanas de tiempo mediante el comportamiento de las hormigas en un hormiguero artificial, teniendo como objetivo minimizar una función objetivo jerárquica simple: el primer objetivo es minimizar el número de recorridos (o vehículos) y el segundo es minimizar el tiempo total de viaje.

1.4. Comparación del desempeño computacional de los algoritmos genético de Chu-Beasley y Colonia de Hormigas en la Solución del problema de P-Centdiana.

Autores: Cesar Adrián Muñoz, Ramon Gallego, Eliana Mirledy Toro.

Descripción: La P-Centdiana es un problema de localización de gran aplicabilidad en la vida diaria que constituye una importante línea de investigación. Entre las aplicaciones tienen: Localización de entidades de servicio, antenas repetidoras de telecomunicaciones, transformadores en sistemas de distribución de energía eléctrica, equipos de comunicaciones para el comando a distancia de dispositivos de control y maniobra entre otros. En este trabajo

se presenta el problema general de la P-Centdiana, así como el modelo matemático que lo representa y la codificación, así como también se describen las técnicas de optimización combinatorial implementadas (Algoritmo Genético de Chu-Beasley y Algoritmo de Colonia de Hormigas) a partir de las cuales se resuelve dicho problema, finalmente se analiza el desempeño de ambos métodos.

Aportes a la tesis: El artículo muestra un análisis útil para entender el desempeño computacional que tienen los métodos a implementar en este trabajo investigativo, de tal manera que se evidencia que el algoritmo genético de Chu-Beasley presenta un mejor desempeño computacional comparado con el algoritmo de Colonia de hormigas, encontrando soluciones de buena calidad. Los métodos se comparan a partir de un caso de prueba, en el cual la información de los posibles centros y los puntos de consumo son generados de forma aleatoria a partir de la distribución de probabilidad uniforme, el Caso consiste en 600 nodos de consumo y 200 centros de servicio candidatos, para el cual se debe ubicar de forma óptima un número de centros de servicios P definidos previamente y basados en algún criterio, por ejemplo, un criterio económico.

1.5. Optimización de un ruteo vehicular usando algoritmo genético simple Chu-Beasley

Autores: Guillermo Roberto Solarte Martínez, Andrés Gaspar Castillo Sanz, Guillermo Rodríguez Gahona

Descripción: En este artículo se realiza una investigación del problema de ruteo CVRP utilizando el algoritmo Chu Beasley se propuso un estudio para aplicar esta metaheurístico en Bogotá, ya que por ser la capital de Colombia tiene dificultad de movilidad y transporte, y se toma como caso de estudio los Súper almacenes Olímpica (SAO) de la ciudad, que deben comercializar sus productos y se ven obligados a optimizar recorridos de entrega para garantizar la satisfacción del cliente.

Aportes a la tesis: Este articulo resulta ser útil para la comprensión del algoritmo genético de Chu-Beasley aplicado a un problema de ruteo con capacidad. Como resultado de esta investigación se desarrolló una aplicación que generó resultados aceptables, reduciendo notablemente los costos de recorrido; mediante esta investigación los autores proponen una solución al problema de ruteo mediante el algoritmo Chu Beasly teniendo en cuenta 14 bodegas en la ciudad de Bogotá y midiendo las distancias entre dos puntos usando la herramienta “Distance Measurement Tool” de Google, además de que se considera que para este caso la distancia entre [AB] es diferente a [BA], por esta razón, el número de combinaciones posibles para este caso es dado por n, donde n es el número de posibles sedes. De esta manera, optimizando tiempos de distribución y entrega, aplicando las fases de los algoritmos genéticos.

1.6. Solución al problema de ruteo de vehículos con capacidad limitada CVRP usando una técnica metaheurística.

Autores: Juan Pablo Orrego Cardozo, Daniela Ospina Toro, Eliana Mirledy Toro Ocampo

Descripción: Los autores propone en este articulo generar una solución para el CVRP tradicional usando la heurística de barridos de dos fases y resolviendo cada problema del agente viajero con el algoritmo genético modificado de Chu-Beasly, la primera parte del algoritmo se plantea bajo la necesidad de presentar una metodología de solución basada en la heurística constructiva de barrido de dos fases, donde inicialmente se agrupen un conjunto de clientes que serán satisfechos por un camión dado y luego se generen las rutas optimas a través del algoritmo genético de Chu Beasly (AGCB) para cada vehículo, esto con el fin de minimizar los costos de recorrido para todas las rutas mínimas necesarias.

Aportes a la tesis: Este artículo proporciona información sobre el comportamiento de un problema de ruteo con capacidad restringida, en donde cada vehículo tiene una capacidad C, y son requeridos para satisfacer a unos n clientes, además de que se muestran distintas técnicas de solución aproximadas para el CVRP, donde las que se usan frecuentemente son las heurísticas y las metaheurísticas, con las cuales se pueden obtener buenas soluciones en tiempos de cómputo razonables; haciendo énfasis en la técnica de dos fases, la cual parte de un problema y una solución "vacía" para que a partir de ella se pueda construir una solución factible, La heurística de dos fases dividen el problema VRP en dos etapas: una de asignación de clientes a vehículos y luego otra para la determinación del orden de visita a dichos clientes.

1.7. Una metaheurística híbrida aplicada a un problema de planificación de rutas.

Autores: Daniel Soto, Wilson Soto, Yoan Pinzón.

Descripción: En este articulo los autores proponen un algoritmo híbrido entre un algoritmo genético y un algoritmo de colonia de hormigas para tratar el problema de recolección básico con una flota de capacidad homogénea, múltiples depósitos y un periodo de m días. Finalmente, este trabajo muestra experimentalmente, el comportamiento del algoritmo híbrido en encontrar una solución óptima para el problema particular de recolección, el algoritmo hibrido desarrollado está compuesto de un algoritmo genético como clasificador y un algoritmo de colonia de hormigas como optimizador local de cada subruta.

Aportes a la tesis: Este articulo proporciona información valiosa sobre el funcionamiento de colonia de hormigas y ayuda a comprender y ampliar la base del conocimiento para el correcto desarrollo del modelo hibrido propuesto en este trabajo, dentro del artículo se desarrolla un algoritmo genético con cada una de las fases que este conlleva para después se recibidos por colonia de hormigas para ser optimizados localmente y tratar de obtener la mejor respuesta de la distribución obtenida con el algoritmo genético. Cada cromosoma de la población es enviado uno por uno como un vector al que se le aplica el algoritmo de colonia de hormigas.

1.8. CVRPTW model applied to the collection of food donations.

Autores: Iván Guillermo Peña Arenas, Alexander Gutiérrez Sánchez, Carlos Armando, López Solano, Linda Bibiana Rocha Medina.

Descripción: El documento muestra un problema de enrutamiento de vehículos capacitados con ventanas de tiempo (CVRTPW) aplicado a la recolección de donaciones de un banco de alimentos. El objetivo principal es minimizar los costos y los tiempos de operación que se ven afectados por el hecho de estar sujetos a una variación dinámica en la programación de donantes para visitar. El problema se resuelve mediante un modelo de programación lineal de enteros y la matriz de distancia calculada a través de la API de Google Maps; Este problema tiene diferentes características, primero, todos los donantes tienen diferentes cantidades de donación y cada uno de ellos tiene su propia ventana de tiempo. La flota de vehículos es homogénea y tiene un único depósito, El banco de alimentos, donde todos los vehículos parten y regresan después de la recolección de las donaciones.

Aportes a la tesis: El artículo genera un aporte a la tesis ya que presenta una solución de programación lineal, donde a través de un servidor externo de Google se conecta con el software local que tiene programación entera mixta y distintos componentes, lo cual ayuda a entender cómo interpretar las distancias obtenidas para un problema de enrutamiento con ventanas de tiempo.

1.9. Ruteo de Vehículos en el sector de hidrocarburos aplicando colonia de hormigas.

Autores: Andrés Felipe Castiblanco Suárez, Daniel Felipe Martin Vargas.

Descripción: Los autores proponen una solución para un problema de ruteo vehicular en el sector de hidrocarburos a través de la metaheurística de Colonia de hormigas, trabajando con una flota homogénea de 200 barriles, además, se tienen en cuenta los tiempos de servicio al cliente (carga y descarga) y ventanas de tiempo. El objetivo es disminuir el número de vehículos que se contratan, determinando la ruta para cada uno de ellos, además de garantizar el cumplimiento de las demandas durante los límites de tiempo establecidos.

Aportes a la tesis: El documento aporta una forma de desarrollar el modelo matemático, con el fin de definir los conjuntos, variables de decisión, parámetros, función objetivo y restricciones asociados al problema a desarrollar y sirve de guía para realizar la metaheurística ACO.

1.10. Un híbrido de algoritmos de colonia de hormigas y luciérnagas (HAFA) para resolver problemas de enrutamiento de vehículos.

Autores: Rajeev Goe, Raman Maini.

Descripción: En el presente artículo se desarrolla un algoritmo hibrido HAFA que incorpora ciertos aspectos de la optimización de luciérnagas y el algoritmo de colonia de hormigas, El rendimiento del algoritmo propuesto se compara con

algunos de otros enfoques metaheurísticos existentes mediante pruebas en ciertos conjuntos de datos de referencia estándar.

Aportes a la tesis: Esta investigación ofrece una visión para desarrollar un modelo híbrido utilizando la metaheurística de colonia de hormigas para resolver problemas de ruteo vehicular. Además, aporta un nuevo procedimiento de agitación con feromonas para evitar quedar atrapado fácilmente en óptimos locales evitando el estancamiento de feromonas en las regiones explotadas. Plantea un modelo matemático cuya estrategia de solución se centra en la minimización del número de rutas (es decir, los vehículos utilizados) como objetivo principal y la distancia total recorrida (o tiempo) secundariamente.

1.11. Solving Vehicle Routing Problem using Ant Colony Optimization (ACO) Algorithm.

Autores: Wan Amir Fuad Wajdi Othman, Aeizaal Azman Wahab, Syed Sahal Nazli Alhady, Haw Ngie Wong.

Descripción: El estudio trata de resolver el problema de enrutamiento de vehículos (VRP) utilizando el algoritmo de optimización de colonias de hormigas (ACO). Se explican las dos fases de ACO: la construcción de rutas de hormigas y la actualización de feromonas. Para esta última se utiliza la actualización local de feromonas. Los resultados de la ruta de diferentes iteraciones se compararon y analizaron el rendimiento del algoritmo. El mejor conjunto de parámetros de control obtenidos es con 20 hormigas, $\alpha = 1$, $\beta = 1$ y $\rho = 0.05$. También se evaluó el costo promedio y la desviación estándar de

los 20 tiempos de ejecución con el mejor conjunto de parámetros de control, con 1057.839 km y 25.913 km respectivamente.

Aportes a la tesis: Provee una metodología para probar la combinación de diferentes parámetros de control de ACO y así encontrar la mejor combinación de los parámetros de control (número de hormigas(nAnt), Alpha(α), beta(β), rho(ρ)) donde Alfa representa la importancia relativa del rastro, beta representa la importancia de la visibilidad y rho representa el parámetro que rige la descomposición de las feromonas.

1.12. Un algoritmo hibrido de optimización de colonia de hormigas para un problema de enrutamiento de vehículos de objetivos múltiples con ventanas de tiempo flexible.

Autores: Huizhen Zang, Qinwan Zhang, Liang Ma, Ziying Zhnag, Yun Liu.

Descripción: En este documento los costos totales de distribución (incluidos los costos de viaje y los costos fijos del vehículo) vehículo) se minimizan y la satisfacción general del cliente se maximiza. Para lograrlo proponen una estrategia de solución basada en la optimización de colonias de hormigas y tres operadores de mutación (el operador de intercambio, el operador de desplazamiento y el operador inverso). Las operaciones de mutación en el algoritmo ACO se utilizan para diversificar la población y evitar la convergencia a un óptimo local. El objetivo es alterar algunos clientes y producir una nueva solución.

Para elegir valores de parámetros siguen el enfoque donde los valores de parámetros (buenos) se establecen antes de la ejecución de un algoritmo evolutivo dado.

Finalmente, para validar el modelo desarrollan una aplicación de un caso real para la logística de una empresa líder en el sector de las hortalizas y distribución de alimento en china. El problema consiste en 16 puntos de entrega (16 supermercados) servidos directamente desde un depósito (base de suministro de vegetales y alimentos).

Aportes a la tesis: Este artículo provee distintos operadores de mutación que pueden ser usados con la optimización de colonia de hormigas y así generar nuevas soluciones teniendo en cuenta las múltiples ventanas de tiempo, para este articulo al tener ventanas de tiempo flexible aumenta la complejidad computacional ya que esto permite que los vehículos puedan llegar más temprano al cliente que visitan y esto altera a su vez los tiempos de todos los demás nodos que cuentan también con una ventana de tiempo.

MARCO TEORICO

2. ALGORITMOS GENETICOS

2.1. INTRODUCCION

Los algoritmos genéticos (AGs) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el

proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin (1859). Por imitación de este proceso, los Algoritmos genéticos son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de estas.

Los algoritmos genéticos usan una analogía directa con el comportamiento natural. Trabajan con una población de individuos, cada uno de los cuales representa una solución factible a un problema dado. A cada individuo se le asigna un valor o puntuación, relacionado con la bondad de dicha solución. En la naturaleza esto equivaldría al grado de efectividad de un organismo para competir por unos determinados recursos. Cuanto mayor sea la adaptación de un individuo al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material genético con otro individuo seleccionado de igual forma. Este cruce producirá nuevos individuos descendientes de los anteriores – los cuales comparten algunas de las características de sus padres. Cuanto menor sea la adaptación de un individuo, menor será la probabilidad de que dicho individuo sea seleccionado para la reproducción, y por tanto de que su material genético se propague en sucesivas generaciones.

2.2. HISTORIA

Entre la década de los 50 y los 80 comenzaron distintos estudios sobre la selección natural, en el 1957 Alex Fraser publicó un artículo referente al tema, gracias a esto; se da inicio a la simulación por computadores de la evolución biológica. En 1975, John Holland publicó su libro "Adaptación en Sistemas Naturales y Artificiales". Basado en investigaciones y artículos anteriores del propio Holland y de colegas de la Universidad de Michigan, donde presentó sistemática y rigurosamente el concepto de sistemas digitales adaptativos utilizando la mutación, la selección y el cruzamiento, simulando el proceso de la evolución biológica como estrategia para resolver problemas. Ese mismo año, la importante tesis de Kenneth De Jong estableció el potencial de los AGs demostrando que podían desenvolverse bien en una gran variedad de funciones de prueba (Goldberg 1989).

En la década de los 80, los algoritmos genéticos se estaban aplicando en una amplia variedad de áreas, desde problemas matemáticos abstractos hasta asuntos tangibles de ingeniería como el control de flujo en una línea de ensamble, reconocimiento y clasificación de patrones y optimización estructural (Goldberg 1989).

Al principio, estas aplicaciones eran teóricas. Sin embargo, al seguir incrementando la investigación, los algoritmos genéticos migraron hacia el sector comercial, al cobrar importancia con el crecimiento exponencial de la potencia de computación y el desarrollo de Internet. Y en el corazón de todo esto se halla nada más que la simple y poderosa idea de Charles Darwin mencionada al principio: que el azar en la variación, junto con la ley de

la selección, es una técnica de resolución de problemas de inmenso poder y de aplicación casi ilimitada.

2.3. DEFINICION

Los algoritmos genéticos son una técnica de búsqueda basada en la teoría de la evolución de Charles Darwin, usada como estrategia para resolver problemas. El poder de los algoritmos genéticos proviene del hecho de que se trata de una técnica robusta, y pueden tratar con éxito una gran variedad de problemas provenientes de diferentes áreas, incluyendo aquellos en los que otros métodos encuentran dificultades. Si bien no se garantiza que el algoritmo genético encuentre la solución óptima del problema, existe evidencia empírica de que se encuentran soluciones de un nivel aceptable, en un tiempo competitivo con el resto de los algoritmos de optimización combinatoria En el caso de que existan técnicas especializadas para resolver un determinado problema, lo más probable es que superen al algoritmo genético, tanto en rapidez como en eficacia. El gran campo de aplicación de los algoritmos genéticos se relaciona con aquellos problemas para los cuales no existen técnicas especializadas. Incluso en el caso en que dichas técnicas existan, y funcionen bien, puede efectuarse mejoras de las mismas hibridándolas con los algoritmos genéticos.

Su funcionamiento parte de una población inicial elegida mediante un proceso de selección, muchas veces se escogen los mejores adaptados para después reproducirlos, mutarlos y recombinarlos mediante los operadores genéticos, de esta manera se puede garantizar que la nueva generación de individuos tenga mejor adaptabilidad.

2.4. CODIFICACIÓN

La codificación de los algoritmos genéticos se representa mediante una cadena de valores conocida como cromosoma, computacionalmente se habla de una cadena de bits en donde cada gen puede ser codificado por un sistema tipo número o alfabético, desde el inicio de los estudios de los algoritmos genéticos se usa la codificación de [0,1] para los alelos por cada gen. La figura 1 muestra la representación de los conceptos mencionados anteriormente.

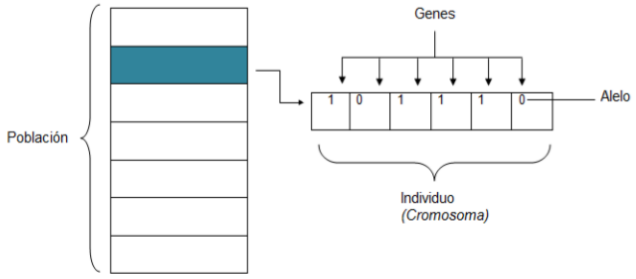

Figura 1. Componentes de un Algoritmo Genético.

Fuente: Arias Claudia (2015). ANÁLISIS E IMPLEMENTACIÓN DEL ALGORTIMO GENÉTICO DE CHUBEASLEY PARA RESOLVER EL PROBLEMA DEL AGENTE VIAJERO (TSP) Y SU VARIANTE, EL PROBLEMA DE RUTAS DE VEHÍCULO (VRP)

2.5. OPERADORES GENETICOS

Dentro de las investigaciones y estudios que se han llevado a cabo se han considerado como fundamentales tres operadores genéticos l*a selección, cruce y la mutación.* Estos permiten que los cromosomas puedan cambiar y evolucionar para encontrar otras soluciones óptimas y garantizar la supervivencia de la población en las siguientes generaciones.

2.5.1. OPERADOR GENETICO DE SELECCIÓN

El proceso de selección sirve para escoger a los individuos de la población mejor adaptados, para que actúen de progenitores de la siguiente generación. En la naturaleza existen varios factores que intervienen para que un individuo pueda tener descendencia. El primero de todos es que consiga sobrevivir, ya sea porque no es devorado por depredadores, o porque sea capaz de procurarse alimento. Lo segundo es que encuentre pareja para reproducirse. El último factor es que la combinación de ambos individuos sea apta para crear un nuevo individuo. Sin embargo, en la realidad es posible que "el mejor" individuo no pueda reproducirse, pero otro individuo de "peor calidad" pueda conseguirlo. Aunque este hecho es menos probable, sigue siendo posible.

En los algoritmos genéticos, la selección es un conjunto de reglas que sirven para elegir a los progenitores de la siguiente generación. Estos progenitores se reproducirán (cruzamiento genético) y generarán descendencia; al ser fundamental existen diferentes métodos de selección, dentro de los que se destacan los siguientes:

Selección por Ruleta

Este tipo selección se emplea mediante una ruleta dividida en sectores proporcionales a la función objetivo, los individuos mejores (con mayor funcion objetivo) son los que tienen más posibilidades de ser elegidos en el proceso.

Selección por torneo

Este sistema K/L consiste en seleccionan K individuos de la población aleatoriamente y de estos K individuos se seleccionan los L que tengan mejor función objetivo. Es uno de los métodos más usados para el proceso de selección.

Selección por Ranqueo

Este tipo de selección los individuos son ordenados de acuerdo a su función objetivo. De esta manera, si tenemos n cromosomas, el individuo con peor función objetivo se le asignará un 1 y el que tenga la mejor función se le asignará la n.

Otros métodos de selección que se emplean son:

- Selección Jerárquica
- Selección por estado estacionario
- Selección escalada
- Selección elitista

2.5.2. OPERADOR GENETICO DE CRUCE

El operador de cruce es el encargado de generar nuevos individuos dentro de los algoritmos genéticos cruzando los cromosomas de los individuos progenitores. Dentro de los tipos de operadores de cruce existentes son:

Cruce simple o cruce de un solo punto

Este cruce consiste en elegir un punto aleatorio sobre dos individuos padres y después realizar la división de cada padre en el punto de cruce elegido. La información genética de uno de los padres desde la parte inicial hasta el punto

de cruce se copia y el resto de información es copiada de la parte restante del otro padre. Un ejemplo de esto se muestra en la figura 2.

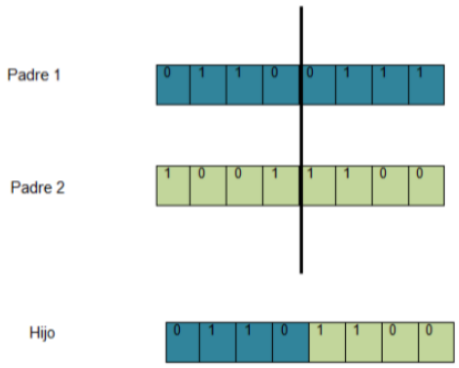

Figura 2. Cruce de un solo punto

Fuente: Arias Claudia (2015). ANÁLISIS E IMPLEMENTACIÓN DEL ALGORTIMO GENÉTICO DE CHUBEASLEY PARA RESOLVER EL PROBLEMA DEL AGENTE VIAJERO (TSP) Y SU VARIANTE, EL PROBLEMA DE RUTAS DE VEHÍCULO (VRP)

Cruce de dos puntos

El cruce de dos puntos es similar al anterior, con la diferencia de que se seleccionan aleatoriamente

2.5.3. OPERADOR GENETICO DE MUTACION

La mutación modifica ciertos genes de forma aleatoria, la cual depende de la codificación del problema y de la selección.

El proceso de mutación se puede realizar de diversas formas, dos de las cuales son:

- *Inversión de genes,* en donde se seleccionan los genes de manera aleatoria y se invierte su valor, en el caso de codificación numérica se sustituye un numero por otro, por ejemplo, se cambia un 0 por un 1 o viceversa.
- *Cambio de orden,* en el cual se seleccionan dos genes de manera aleatoria y se realiza el intercambio de posiciones.

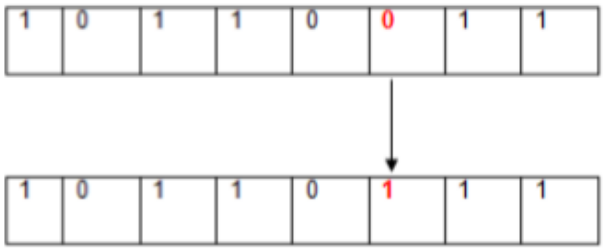

Figura 3. Mutación con inversión de genes.

Fuente: Arias Claudia (2015). ANÁLISIS E IMPLEMENTACIÓN DEL ALGORTIMO GENÉTICO DE CHUBEASLEY PARA RESOLVER EL PROBLEMA DEL AGENTE VIAJERO (TSP) Y SU VARIANTE, EL PROBLEMA DE RUTAS DE VEHÍCULO (VRP)

3. PROBLEMA DE RUTEO DE VEHICULOS (VRP)

3.1. INTRODUCCION

En las últimas décadas ha habido un incremento de paquetes de optimización basados en técnicas de investigación de operaciones o programación matemática, en sistemas de distribución para el manejo efectivo de la provisión de bienes o servicios. De acuerdo a aplicaciones del mundo real se ha mostrado que

una buena planeación de los procesos de distribución genera ahorros del 5% al 20% en los costos de transportación global, es de suma importancia establecer un modelo que permita mejorar el desempeño logístico, pues como lo expresaron Toth y Vigo (2002) *"El problema de distribuir productos desde ciertos depósitos a sus usuarios finales juega un papel central en la gestión de algunos sistemas logísticos, y su adecuada planificación puede significar considerables ahorros, Esos potenciales ahorros justifican en gran medida la utilización de técnicas de investigación operativa como facilitadoras de planificación, dado que se estima que los costos del transporte representan entre el 10% y el 20% del costo final de los bienes".* De acuerdo con lo anterior, las investigaciones y estudios llevaron al planteamiento del problema de ruteo de vehículos (VRP en inglés, Vehicle Routing Problem).

> *"El problema de ruteo de vehículos es conocido como uno de los problemas de optimización combinatoria que pertenece a la clase NP-Hard, dado que no es posible resolverlos en un tiempo polinómico; su planteamiento se basa en encontrar un conjunto de rutas a un costo mínimo, la cual inicia y finaliza en el depósito, cumpliendo con la demanda de cada uno de los clientes a los cuales debe visitar."*(Arias,2015)

3.2. HISTORIA

El surgimiento del problema de ruteo de vehículos (VRP) se remonta al año 1956 cuando a su vez se planteaba el problema del agente viajero, Dantzing y Ramser en su libro *"The truck dispatching problem"*(1959) se planteó un modelo

de combustible que se realizaba a través de una de una flota de camiones a distintas estaciones de servicio.

En el año de 1981 Lenstra y Rinnooy Kan, analizaron la complejidad del problema de ruteo de vehículos, en donde concluyeron que estos problemas pertenecen a la clase NP-Hard, dado que no se pueden resolver en tiempo polinomial.

3.3. DEFINICION

El problema de enrutamiento de vehículos puede describirse como el problema de diseñar rutas optimas de entrega o recolección de uno o varios depósitos a un numero de ciudades o clientes geográficamente dispersos, los cuales están sujetos a restricciones, El VRP es un problema de programación de optimización combinatoria y entera que realiza la siguiente pregunta "¿Cuál es el conjunto óptimo de rutas que una flota de vehículos debe recorrer para entregar a un conjunto dado de clientes?".

El problema consiste en que una flota de vehículos debe visitar un conjunto de ciudades o clientes, teniendo un depósito central que es el punto de partida de los vehículos, se tiene como objetivo poder encontrar un conjunto de rutas que minimicen los costos del recorrido, visitando cada cliente una sola vez y con un solo vehículo.

El problema de enrutamiento de vehículos está compuesto por clientes, depósitos y vehículos, los cuales según las características puede generar variantes de este problema.

- Los clientes o nodos que deben ser visitados por lo menos una vez por los vehículos, estos tienen diferentes variables como la demanda, el tiempo de servicio, tiempo, horarios y una serie de restricciones para los vehículos.
- Los depósitos que pueden ser uno o varios dependiendo del problema que se plantea, este generalmente se encuentra ubicado en una zona central o estratégica para garantizar la mejor distribución de la rutas que deben visitar a cada cliente, desde allí parten los vehículos y se especifica de manera general que cada ruta inicie y termine en un mismo periodo.
- Los vehículos que también poseen varias variables como lo pueden ser la capacidad y el costo (variables y fijos), se pueden clasificar en una flota homogénea (Todos los vehículos son iguales en características) o flota heterogénea (Poseen características diferentes).

En la figura 4 se observa la representación del VRP y los componentes descritos anteriormente.

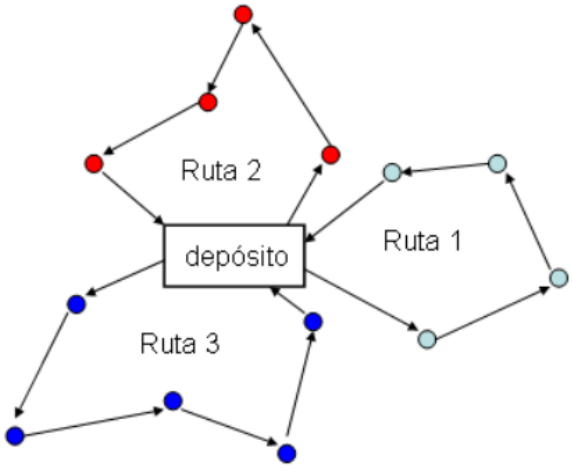

Figura 4. Representación del Problema de enrutamiento vehicular.

Fuente: (Calvillo,2010). El problema de ruteo de vehículos. Universidad Autónoma de Coahuila.

3.4. TIPOS DE VRP

El problema de enrutamiento de vehículos al ser objeto de estudio durante muchos años ha tenido un amplio campo de aplicación, por sus características es posible implementarlo de muchas formas dependiendo de las necesidades o restricciones que el problema demande, debido a esto, el VRP ha tenido varios enfoques que permiten una mejor búsqueda de soluciones con una toma de decisiones mucho más precisa. En la figura 5 se muestran los distintos tipos de VRP

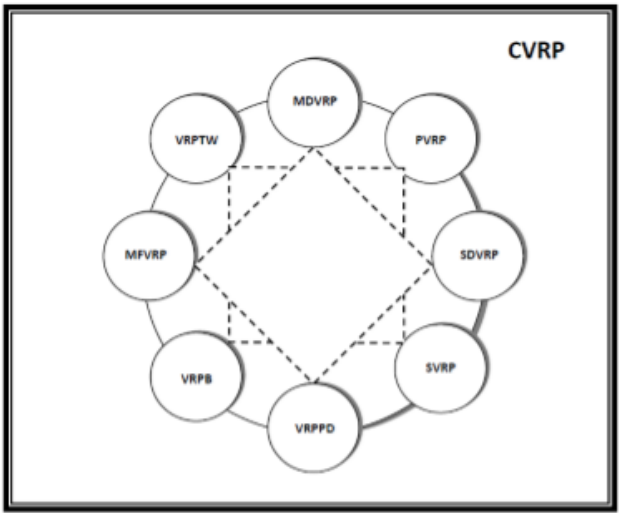

Figura 5 Variantes del VRP.

Fuente: Arias Claudia (2015). ANÁLISIS E IMPLEMENTACIÓN DEL ALGORTIMO GENÉTICO DE CHUBEASLEY PARA RESOLVER EL PROBLEMA DEL AGENTE VIAJERO (TSP) Y SU VARIANTE, EL PROBLEMA DE RUTAS DE VEHÍCULO (VRP)

Problema con restricciones de capacidad (CVRP)

El problema CVRP (Capacited VRP) es una variante del VRP que tiene como restricción principal la capacidad que tiene la flota de vehículos y un único deposito, el objetivo es minimizar la distancia total recorrida y se satisfaga la demanda de los clientes sin exceder la capacidad de los vehículos.

Problema con múltiples depósitos (MDVRP)

El problema con múltiples depósitos conocido como MDVRP (por sus siglas en inglés, Multiple Depot VRP), tiene el mismo objetivo que el VRP tradicional con la diferencia de que existen varios depósitos desde donde se puede ofrecer el servicio a los clientes, cada uno dispone de su propia flota de vehiculos.

Problema periódico (PVRP)

El PVRP (Periodic VRP) consiste en establecer un horizonte de operación M días, en este periodo se deben visitar al menos una vez a cada cliente.

Problema de entregas parciales (SDVRP)

El problema SDVRP (Split Delivery VRP) o también conocido como VRP de entrega dividida por diferentes vehículos reduciendo los costos, siempre que la demanda del cliente sea mayor a la capacidad de los vehículos.

Problemas con valores al azar (SVRP)

El problema SVRP (Stochastic SVRP) pretende minimizar la distancia recorrida que a partir de la aleatoriedad de las variables se pueda cumplir con sus demandas, servicios y tiempos de viaje.

Problema con entregas y devoluciones (VRPPD)

El problema VRPPD (VRP With Pickup and Delivery) plantea que existe la posibilidad que los clientes puedan devolver algún tipo de mercancía, por lo que se deben tener en cuenta que estos artículos si quepan en el vehículo y no excedan su capacidad. El objetivo de este problema consiste en minimizar la flota y la suma de los tiempos de transporte bajo la restricción de que cada uno de los vehículos debe contar con la capacidad suficiente para transportar los productos que deberán recogerles a los clientes con el fin de regresarlos al depósito.

Problemas con viajes de regreso (VRPB)

El problema VRPB (VRP With Bachauls) tiene un planteamiento similar al VRPPD que consiste en que los clientes pueden tanto recibir mercancías o productos como entregarlos, pero este tiene la variante de que existe una restricción en la que las entregas deben ser realizadas y completadas antes de que se realicen las devoluciones.

Problema con múltiples depósitos (MFVRP)

El problema MFVRP (Mixed Fleet VRP) consiste en que los vehículos pueden tener distintas capacidades, por lo cual se deberá determinar que vehículo será el adecuado para realizar el recorrido de una ruta según la distancia y la demanda.

Problema con ventanas de tiempo (VRPTW)

El problema VRPTW (VRP With time Windows) tiene el mismo planteamiento del VRP, pero con la restricción de que existe un periodo de tiempo determinado para cumplir con el abastecimiento de los clientes.

4. FORMULACION DEL PROBLEMA

El problema de enrutamiento de vehículos con ventanas de tiempo (VRPTW) es dado por una flota homogénea de vehículos denotados V,un conjunto de clientes C y un grafo dirigido $G = (V, C)$. El grafo consiste en $|C| + 1$ vértices, donde los clientes son denotados como 1, 2, …, n y el depósito es representado por el vértice 0.

El VRPTW tiene múltiples objetivos en donde la meta es minimizar no solamente el número de vehículos requeridos, sino también el tiempo total de viaje, el tiempo de espera y la distancia total de viaje incurrida por la flota de vehículos. El conjunto de arcos denotado por A representa las conexiones entre el depósito y los clientes y entre los clientes. Con cada arco (i, j), donde $i \neq j$, se le asocia un costo c_{ij} y un tiempo t_{ij} el cual puede incluir un tiempo de servicio al cliente i.

Cada cliente i tiene una ventana de tiempo $[a_i, b_i]$. Un vehículo debe llegar al cliente antes de que se cumpla el tiempo b_i y después de a_i, pero no puede llegar antes a realizar el servicio.

Se asume que a_i, b_i, c_{ij} son enteros no negativos, mientras que el t_{ij} se asume que es un entero positivo, también se tiene en cuenta que la inecuación triangular se satisface tanto para c_{ij} como para t_{ij}. El modelo contiene dos

conjuntos de variables de decisión x_{ijk} y s_{ik}. Para cada arco (i, j) donde $i \neq j$, $i \neq 0$, $j \neq 0$, y para cada k se define que $x_{ijk} = 1$ se dice que la solución es óptima, de lo contrario, el arco (i, j) es atravesado por un vehículo k e igual a 0.

La variable de decisión s_{ik} es definida para cada vértice i y cada vehículo k y denota el tiempo en que el vehículo k inicia el servicio en el cliente i, En el caso en que el vehículo dado no realiza ningún servicio al cliente i, s_{ik} no significa nada, Se asume que $a_0 = 0$ por lo tanto $s_{0k} = 0$, para todo k. Se quiere diseñar un conjunto de rutas al mínimo costo, una para cada vehículo, de modo que cada cliente sea visitado exactamente una vez, en donde cada ruta se origine en el vértice 0 (El depósito) y finalice en el mismo, satisfaciendo las ventanas de tiempo de cada cliente. En la figura 6 se observa el modelo matemático para el VRPTW con sus respectivas restricciones.

$$\sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{jk} \qquad (2.1)$$

$$\sum_{k \in v} \sum_{j \in N} x_{ijk} = 1 \quad \forall i \in C \qquad (2.2)$$

$$\sum_{i \in N} x_{0jk} = 1, \quad \forall k \in V \qquad (2.3)$$

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0, \quad \forall h \in C, \quad \forall k \in V \qquad (2.4)$$

$$\sum_{i \in N} x_{i0k} = 1, \quad \forall k \in V \qquad (2.5)$$

$$s_{ik} + t_{ij} - K\left(1 - x_{ijk}\right) \leq s_{jk}, \quad \forall i, j \in N, \quad \forall k \in V \qquad (2.6)$$

$$a_i \leq s_{ik} \leq b_i, \quad \forall i \in N, \quad \forall k \in V \qquad (2.7)$$

$$x_{ijk} \in \{0,1\}, \forall i, j \in N, \quad \forall k \in V \qquad (2.8)$$

La restricción (2.2) establece que cada cliente es visitado exactamente una vez. Las siguientes tres ecuaciones (2.3) ,(2.4) y (2.5) aseguran que cada vehículo abandone el depósito 0, después de llegar a un cliente el vehículo se va de nuevo y finalmente llega al depósito. Las desigualdades (2.6) establecen que cada vehículo k no puede llegar a j antes que $s_{ik} + t_{ij}$ si está viajando de i a j. Aquí K es un número muy grande. Finalmente, la restricción (2.7) asegura que las ventanas de tiempo son cumplidas y la (2.8) que son las restricciones de integralidad.

5. ALGORITMO GENETICO DE CHU-BEASLEY

5.1. INTRODUCCIÓN

El algoritmo Chu-Beasley es una versión modificada del algoritmo genético básico, la principal característica consiste en mantener diversidad entre los cromosomas que conforman la población durante todo el proceso. En cada generación es reemplazado un solo cromosoma (alternativa) en la población, siempre y cuando, cumpla con las condiciones de optimalidad y/o factibilidad establecidas. Dicho mecanismo busca que, en cada ciclo generacional, la calidad de la solución sea mejorada por optimalidad y/o factibilidad. Durante el proceso en la población se reemplaza sistemáticamente un único descendiente. Esta estrategia tiene como ventaja encontrar soluciones de alta calidad y garantizar diversidad en la población a lo largo de las generaciones.

El algoritmo genético de Chu-Beasley (AGCB) es capaz de generar soluciones a problemas de optimización en donde las técnicas exactas no son eficientes, esto gracias a que pertenece al grupo de las metaheurísticas y que tiene una fase de mejora local de la descendencia de los padres que permite escoger a los mejor adaptados para reemplazar a los peores dentro de toda la población.

5.2. DEFINICION

El AGCB es una técnica metaheurística que utiliza una serie de funciones propias de un algoritmo genético usando la función fitness. El algoritmo genético de Chu-Beasley es una "versión modificada" de los algoritmos genéticos tradicionales y presenta modificaciones y características particulares como:

1. Utiliza la función objetivo para identificar el valor de la solución de mejor calidad y maneja la infactibilidad para el proceso de reemplazo de una solución generada a través del proceso de selección-recombinación-mutación por otra que se encuentra en la población actual.

2. A diferencia del AG propuesto por Holland, el algoritmo AGCB sólo genera y sustituye una configuración a la vez en la población, en cada ciclo generacional.

3. Es un algoritmo elitista, ya que un padre será reemplazado por un descendiente en la próxima generación, si y sólo si, el descendiente tiene una función de ajuste de mejor calidad que el padre.

4. Cada configuración que entra a hacer parte de la población debe ser diferente a todos los que conforman la población actual, lo que evita la convergencia prematura a soluciones óptimas locales.

5. Puede incluir una etapa de mejoramiento después de realizar selección, recombinación y mutación. Esto permite explotar la solución descendiente antes de determinar si puede reemplazar a un individuo de la población actual.

El algoritmo genético de Chu-Beasley realiza los procesos evolutivos llevados a cabo por los algoritmos genéticos tradicionales, como lo son la selección de los cromosomas padres, La recombinación de sus genes para generar el nuevo individuo y La mutación de los alelos , por ultimo se agrega un proceso de comparación de los individuos para reemplazarlo por el de peor calidad dentro de la población.

5.3. FUNCIONAMIENTO DEL ALGORITMO

El algoritmo genético de Chu-Beasley (AGCB) tiene un funcionamiento similar a los algoritmos genéticos simples, llevando a cabo los mínimos procesos (selección, recombinación y mutación) pero con algunas variantes. Este algoritmo funciona de la siguiente manera:

1. Se especifica las características genéticas del algoritmo, tales como tipo de codificación, montaje inicial de la población, forma de selección, etc.
2. Se recibe la población inicial generada por el algoritmo de Colonia de Hormigas.
3. Se eliminan los individuos repetidos de la población.
4. Se obtiene la función objetivo de cada individuo de la población.
5. Se codifican los individuos tal como se definió en el paso 1.

6. Se obtienen dos individuos padres empleando selección por torneo de la población actual.
7. Se aplica *Edge Recombination* para cruzar los padres seleccionados anteriormente, en donde se obtiene una alternativa *"hijo".*
8. Se obtiene una alternativa modificada aplicando *Mutación* con *inversión de genes.*
9. Para modificar la población se propone la siguiente estrategia:

a. Si la alternativa actual es infactible y a su vez es menos infactible que la peor infactible de la población, entonces reemplazar la peor infactible por la alternativa actual.
b. Si la configuración es factible y existe por lo menos una infactible en la población actual, entonces reemplazar la peor infactible por la alternativa actual.
c. Si la configuración es factible y todas las alternativas de la población actual son factibles, entonces reemplazar la alternativa con peor función objetivo por la alternativa actual. Lo anterior se realiza sólo si la alternativa actual es de mejor calidad que lo peor de la población

10. En caso contrario desechar la alternativa resultante y volver al paso 6.
11. El proceso termina hasta cumplir el criterio de número máximo de iteraciones.
12. Devuelve la solución con la mejor función objetivo de la población.

En la figura 6, se muestra el esquema del funcionamiento del algoritmo genético de Chu-Beasley

Figura 6. Funcionamiento del algoritmo.

Fuente: Arias Claudia (2015). ANÁLISIS E IMPLEMENTACIÓN DEL ALGORTIMO GENÉTICO DE CHUBEASLEY PARA RESOLVER EL PROBLEMA DEL AGENTE VIAJERO (TSP) Y SU VARIANTE, EL PROBLEMA DE RUTAS DE VEHÍCULO (VRP)

6. COLONIA DE HORMIGAS (ANT COLONY SISTEM)

6.1. INTRODUCCION

La observación de la naturaleza ha sido una de las principales fuentes de inspiración para propuesta de nuevos paradigmas computacionales. Así nacieron diversas técnicas de Inteligencia Artificial como: Los Algoritmos Genéticos (Genetic Algorithms), Templado Simulado (Simulated Annealing), Redes Neuronales (Neural Networks), y entre estas técnicas, el sistema basado en Colonia de Hormigas (Ant Colony System).

Resulta realmente interesante analizar como las hormigas buscan su alimento y logran establecer el camino más corto para luego regresar a su nido. Para esto, al moverse una hormiga, deposita una sustancia química denominada *feromona* como una señal odorífera para que las demás puedan seguirla. Las feromonas son un sistema indirecto de comunicación química entre animales de una misma especie, que transmiten información acerca del estado fisiológico, reproductivo y social, así como la edad, el sexo y el parentesco del animal emisor, las cuales son recibidas en el sistema olfativo del animal receptor, quien interpreta esas señales, jugando un papel importante en la organización y la supervivencia de muchas especies.

Al iniciar la búsqueda de alimento, una hormiga aislada se mueve a ciegas, es decir, sin ninguna señal que pueda guiarla, pero las que le siguen deciden con buena probabilidad seguir el camino con mayor cantidad de feromonas. Considere la figura 9 en donde se observa como las hormigas establecen el camino más corto. En la figura (a) las hormigas llegan a un punto donde tienen que decidir por uno de los caminos que se les presenta, lo que resuelven de manera aleatoria. En consecuencia, la mitad de las hormigas se dirigirán hacia un extremo y la otra mitad hacia el otro extremo, como ilustra la figura (b). Como las hormigas se mueven aproximadamente a una velocidad constante, las que eligieron el camino más corto alcanzarán el otro extremo más rápido que las que tomaron el camino más largo, quedando depositado la mayor cantidad de feromona por unidad de longitud, como ilustra la figura (c). La mayor densidad de feromonas depositadas en el trayecto más corto hace que este sea más deseable para las siguientes

hormigas y por lo tanto la mayoría elige transitar por él. Considerando que la evaporación de la sustancia química hace que los caminos menos transitados sean cada vez menos deseables y la realimentación positiva en el camino con más feromona, resulta claro que al cabo de su tiempo casi todas las hormigas transiten por el camino más corto.

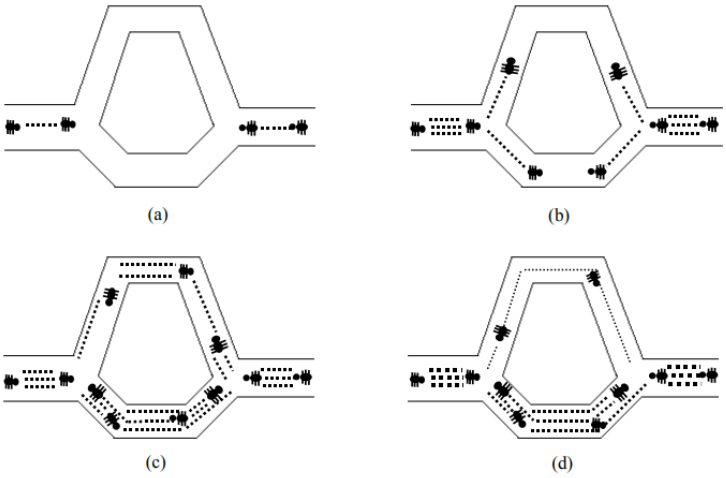

Figura 7. Comportamiento real de las hormigas

Fuente: Barán, B., & Almirón, M. (2001). Colonias distribuidas de hormigas en un entorno paralelo asíncrono.

6.2. DEFINICION

El algoritmo de Optimización por colonia de hormigas es una metaheurística capaz de encontrar soluciones de buena calidad a problemas de optimización altamente complejos. Se basa en la habilidad que poseen las hormigas naturales para encontrar el alimento a partir de individuos relativamente simples, pero con una estructura social altamente eficiente. La base en ambos sistemas (natural y eficiente) es la comunicación indirecta entre todos los

individuos a partir de rastros de feromonas. El algoritmo utiliza agentes muy simples (llamados hormigas) que deben establecer el camino más corto para visitar todas las ciudades del problema una sola vez y regresar a la ciudad origen, para lo cual utilizan la información en una matriz de feromonas.

6.3. FUNCIONAMIENTO DEL ALGORITMO

Se realizaron tres variantes del algoritmo según la forma de realizar el depósito de feromona en el medio:

- *Densidad.* Se realizaba el depósito de feromona durante el transcurso del recorrido (actualización en línea paso a paso de feromona). La cantidad de feromona depositada era siempre constante.
- *Cantidad:* Se realizaba el depósito de feromona durante el transcurso recorrido (actualización en línea paso a paso de feromona). La cantidad de feromona depositada estaba relacionada a la deseabilidad heurística de la trama.
- *Ciclo:* El depósito de feromona se realiza una vez finalizada una solución (actualización en línea a posteriori de feromona).

Esta última variante es la que mejor resultados proporciono y la que se conoce por Ant System (AS) o Sistema de Hormigas.

El AS se caracteriza por el hecho de que la actualización de feromona se realiza una vez que todas las hormigas han completado sus soluciones, y que se lleva a cabo como sigue: primero, rodos los rastros de feromona se reducen en un factor constante, implementadose de esta manera la evaporación de feromona. A continuación, cada hormiga de la colonia deposita una cantidad de feromona que

es función de la calidad de su solución. Inicialmente, el AS no usaba ninguna acción en un segundo plano, pero es relativamente fácil, por ejemplo, añadir un procedimiento de búsqueda local para refinar las soluciones generadas por las hormigas. En la figura 8 se presenta el funcionamiento a grandes rasgos del funcionamiento.

Figura 8. Funcionamiento algoritmo AS.

Fuente: (Atehortúa, 2012). OPTIMIZACION BASADA EN COLONIA DE HORMIGAS: GENERALIDADES Y ESTUDIO DEL ALGORITMO SISTEMA HORMIGA Y APLICACIÓN A UN JOB SHOP

En palabras, el algoritmo es una ejecución continua hasta cumplir una condición de parada, esta condición puede ser establecida de muchas maneras según el objetivo y la disponibilidad de recursos, ejemplos pueden ser un numero de iteraciones especificadas, hasta cuando no se note mejoría en la variable objetivo después de cierto número de iteraciones, etc. En cada ciclo se crea una hormiga que va cambiando de estado o nodo, de acuerdo con una probabilidad resultante de una función heurística y de la cantidad de feromona detectada en ese recorrido concreto *("Mover hormiga").* Se puede decir, que la hormiga es un elemento simple, que se desplaza basándose en información local, tanto heurística, como la aportada por los elementos de la colonia (feromona). Las ciudades no se pueden repetir, por lo que cada hormiga debe tener una lista denominada tabú de las ciudades ya visitadas. Una vez determinada la probabilidad de las diferentes posibles rutas a tomar, la hormiga decide en función a estos valores *("Elección del movimiento*"). Una vez elegido el nodo a visitar, se añade este a la lista de nodos visitados (lista tabú), repitiendo este proceso hasta finalizar la visita de todos los nodos. Una vez terminado un ciclo, se procede a realizar la evaporación de feromona depositada en los arcos de la red y la deposición de feromona sobre la solución obtenida *(Actualización de feromona").* El ciclo re repite hasta la condición de fin establecida para el algoritmo.

Las hormigas utilizan el depósito de feromonas para recordar su comportamiento, es decir, acumular el conocimiento que van adquiriendo del problema a resolver. En un primer momento, todos los arcos presentan la misma probabilidad y para ello se considera oportuno introducir un pequeño valor de

feromona, cantidad que hace posible que caminos, sin explorar también tengan probabilidad de ser recorridos.

7. IMPLEMENTACION DE LA SOLUCION.

La solución para el problema objeto de estudio se hizo con base en un problema de enrutamiento de vehículos con ventanas de tiempo VRPTW, El algoritmo genético y la metaheurística fueron desarrollados en el lenguaje Python y el uso de la API para calcular las distancias proporcionada por Google Maps.

Nota: La flota de vehículos encargada de la recolección de los desechos hospitalarios tiene un total de 4 vehículos recolectores, una distancia total promedio de 195 km y cada ruta un horario establecido de 6:30am de salida y 2:00pm de llegada y finalización al depósito.

Población inicial

Para la población inicial se tuvieron en cuenta un total de 32 Hospitales y Clínicas y un depósito de Salida y Llegada ubicado en el Kilómetro 6 contiguo a Suzuki Cerritos, a partir de ese punto se calculan todas las distancias desde cada uno de los puntos distribuidos por toda el área metropolitana, La hora de inicio para la prueba realizada fue también a las 6:30am para la salida de los vehículos desde el depósito.

Tabla 1. Centros de Salud

	HOSPITAL	DIRECCION		HOSPITAL	DIRECCION
1	Hospital Mental Universitario	Av. 30 de Agosto Cra. 13 # 87 – 76	17	Clínica Comfamiliar	Av. Circunvalar # 3 -01
2	Seccional de Sanidad Ponal	Calle 94 Av. Villa Olímpica	18	ESE – Casa del Abuelo	Cl 63 Bis # 4b – 02 Ciudad Boquía

3	Hospital San Jorge	Cra 4 #24 – 88	19	Hospital San Joaquín	Cra 26 # 78 – 80
4	Clínica San Rafael Cuba	Cra 25 # 74a – 87	20	Centro de Salud Boston	Cra 23 # 20 – 66
5	Clínica Risaralda	Calle 19 # 5 – 13	21	Liga Contra el Cáncer	Cra 4 # 23 – 55
6	Clínica SaludCoop	Cra 7 # 45 – 80	22	Oncólogos de Occidente	Av. Circunvalar # 1 – 46
7	Salud Pereira- Hospital Centro	Cra 7 # 40 – 2	23	Comfamiliar Sede Centro	Cl 30 # 3 – 70
8	Hospital Santa Mónica	Av. Santa Mónica #19a – 18	24	Centro de Salud Villa Santana	Cra 21 # 17e – 49
9	Pinares Medica	Cra 9 # 20 – 60	25	Parque de la Salud	Cl 22 # 20 – 91 La Pradera
10	Clínica los Rosales	Cra 9 # 25 – 25	26	Puesto de Salud Altagracia	Corregimiento Altagracia
11	Fracturas y Fracturas	Cra 12 Bis # 9 – 22	27	Centro Médico Guadalupe	Cra 15 # 35 – 39 Guadalupe
12	Clínica el Lago	Cl 24 # 6 – 26	28	Centro de Salud Santa Teresita	Cra 1 # 17 – 21
13	Fresenius Medical Care	Av. Juan B Gutiérrez # 17 – 55	29	Comfamiliar sede Dosquebradas	Av. Simón Bolívar # 35 – 01
14	MegaCentro Pinares	Cra 18 # 12 – 75	30	Laboratorio Clínico San Rafael	Cra 16 # 12 La Popa
15	Clínica Los Nevados	Cl 20 # 5 – 70	31	IDIME Dosquebradas	Calle 18 18 – 43 Santa Mónica
16	Hospital de Kennedy	Cra 12 Cl 9 Plazuela Estadio Mora Mora	32	Centro de Salud Villa Consota	Cl 70 Cra 30 Esquina de Panorama I

Fuente: Elaboración Propia.

A partir de estos datos se procede a calcular las distancias, para esto se implementa un programa en Python que mediante la conexión a la API de Google Maps se logran extraer, mediante latitud y longitud de cada uno de los nodos, las distancias desde cada uno de los puntos.

```
destination=origins
acu_result=[]
for i in range(0,11):
    result= gmaps.distance_matrix(origins[3*i:3*(i+1)], destination,
mode='driving')
    if i>0:
    acu_result["origin_addresses"]+= result["origin_addresses"]
    acu_result["rows"]+= result["rows"]
```

```
        else:
        acu_result=result

tiempos=[[] for i in range (len(origins))]
distances=[[] for i in range(len(origins))]
for i in range(len(origins)):
for j in range(len(destination)):
      tiempos[i].append(acu_result["rows"][i]["elements"][j]["duration"]["value"]

distances[i].append(acu_result["rows"][i]["elements"][j]["distance"]["value"])

with open('FinalDistancias.txt', 'w') as json_file:
        json.dump(distances, json_file)

with open('FinalTiempos.txt', 'w') as json_file:
        json.dump(tiempos, json_file)
```

Cálculo de distancias API Google Maps.

Fuente: Elaboración Propia

Posterior a este proceso, las distancias calculadas mediante la API de Google Maps son entregadas a Colonia de hormigas, el cual comienza a construir la ruta asignando la misma cantidad de hormigas que nodos, actualizando la feromona cuando termina todas las hormigas han añadido un nodo. A continuación, se presenta una fracción del código de colonia de hormigas.

```
def localUpdate(distancias,visitar,current_time,feromonas,ants,rutas,newclient):
    p=0.01  #tasa de evaporacion
    for ant in range(ants):
        clienteActual= rutas[ant][-1]
        clienteNuevo= newclient[ant]
```

```
        if clienteActual!=clienteNuevo:
            feromonas[clienteActual][clienteNuevo]=(1-
p)*feromonas[clienteActual][clienteNuevo]+p*t0
    rutas[ant].append(clienteNuevo)
    distancias[ant]+= matriz_distance[clienteActual][clienteNuevo]
    if clienteNuevo:
        visitar[ant].remove(clienteNuevo)
        current_time[ant]+= service+matriz_time[clienteActual][clienteNuevo]
    else:
        if visitar[ant]:
            idxmin= tiempos.loc[visitar[ant],"DUE DATE"].idxmin()
            readyTime= tiempos.loc[idxmin,"READYTIME"]
            dueDate = tiempos.loc[idxmin,"DUE DATE"]
            time= matriz_time[0][idxmin]
                current_time[ant]= max(random.randint(readyTime,dueDate)-time,0.0)
    return (distancias,visitar,current_time,feromonas,rutas)
```

Actualización local de feromonas Algoritmo Colonia de Hormigas

Fuente: Elaboración Propia

Una vez que las hormigas hayan terminado de construir cada una de las rutas, estas son enviadas al Algoritmo genético de Chu-Beasley para seleccionar la mejor de todas, para este proceso se envía una por una para comenzar el proceso del algoritmo genético (Selección), realizar cada uno de las etapas (recombinación y mutación) para después comparar el nuevo individuo con el resto de la población y verificar si este es mejor que el peor de las rutas encontradas por colonia de hormigas, mediante la comparación de la función objetivo (Fitness), haciendo el proceso la cantidad de iteraciones necesarias hasta encontrar el punto de convergencia; Para este análisis se hicieron varias pruebas

con distintas cantidades de iteraciones teniendo en cuenta era la misma cantidad tanto para colonia de hormigas como para Chu-Beasley hasta llegar a un máximo de 1000 iteraciones.

```
def GACB(pop,ngen):
    print("Tam pop: "+ str(len(pop)))
    pop=diversity(pop)
    print("Tam diversity: "+ str(len(pop)))
    evaluatePop=list(map(evaluate, pop))
    for gen in range(ngen):
            padre1,padre2=selTournament(pop,evaluatePop)
            child=crossover(flatten(padre1),flatten(padre2))
            child=mutation(child)
            pop,evaluatePop=replace(pop,evaluatePop,child)
    maxFit= max(evaluatePop)
    indxBest= evaluatePop.index(maxFit)
    return pop[indxBest], 1/maxFit
```

Algoritmo Chu-Beasley

Fuente: Elaboración Propia

8. PRUEBAS Y PRESENTACION DE RESULTADOS

Para cada una de las pruebas realizadas se usaron las mismas variables y datos variando la cantidad de iteraciones y la misma cantidad de hormigas, los resultados son presentados a continuación.

El modelo hibrido fue ejecutado en un equipo con las siguientes características:

Marca: ACER.

Modelo: A315-55G-51FD

Procesador: Intel Core i5 8265Ub 1,60 Ghz

Disco duro: 1Tb HHD + 128 GB SSD

RAM: 8GB

Tarjeta de Video: Nvidia Geforce MX230 2G GDDR5

PRUEBA UNO (100 ITERACIONES).

Tabla 2. Datos prueba uno.

100 iteraciones	
Vehículos	3
Mejor Ruta	[[2 1 24 16 15 5 8 25 31 29 28 27], [7 17 22 10 11 12 32 30 6 18 21 23], [19 4 26 3 20 14 13 9]]
Distancia Total (m)	200466
Tiempo de ejecución (Sg)	35.19106388092041
Ruta 1	
Ruta	[2 1 24 16 15 5 8 25 31 29 28 27]
Hora de Inicio (24h)	7:12:48
Hora Fin (24h)	13:36:41
Distancia (m)	64785
Ruta 2	
Ruta	[7 17 22 10 11 12 32 30 6 18 21 23]
Hora de Inicio (24h)	6:37:41
Hora Fin (24h)	13:23:53
Distancia (m)	74060
Ruta 3	
Ruta	[19 4 26 3 20 14 13 9]
Hora de Inicio (24h)	8:54:38
Hora Fin (24h)	13:37:23
Distancia (m)	61621

Fuente: Elaboración Propia

A continuación, se presenta un gráfico en coordenadas de Latitud y Longitud para las rutas creadas en los distintos nodos.

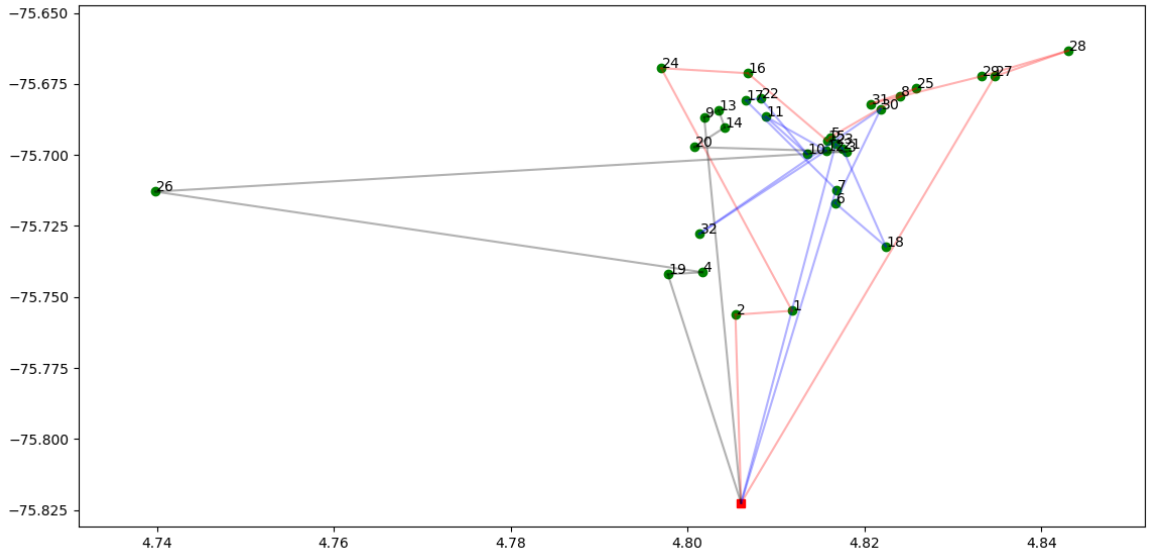

Figura 9. Gráfico Datos prueba uno.

PRUEBA DOS (400 ITERACIONES)

Tabla 3. Datos prueba dos

400 iteraciones	
Vehículos	3
Mejor Ruta	[[2, 1, 10, 12, 4, 32, 30, 8, 25, 31, 21, 23], [7, 17, 22, 11, 24, 16, 15, 5, 20, 14, 3, 9, 13], [19, 26, 6, 18, 27, 29, 28]]
Distancia Total (m)	194660
Tiempo de ejecución (Sg)	136.43349838256836
Ruta 1	
Ruta	[2, 1, 10, 12, 4, 32, 30, 8, 25, 31, 21, 23]
Hora de Inicio (24h)	7:20:58
Hora Fin (24h)	13:29:53

Distancia (m)	61009
Ruta 2	
Ruta	[7, 17, 22, 11, 24, 16, 15, 5, 20, 14, 3, 9, 13]
Hora de Inicio (24h)	6:46:43
Hora Fin (24h)	13:32:54
Distancia (m)	61503
Ruta 3	
Ruta	[19, 26, 6, 18, 27, 29, 28]
Hora de Inicio (24h)	9:07:30
Hora Fin (24h)	13:38:56
Distancia (m)	72147

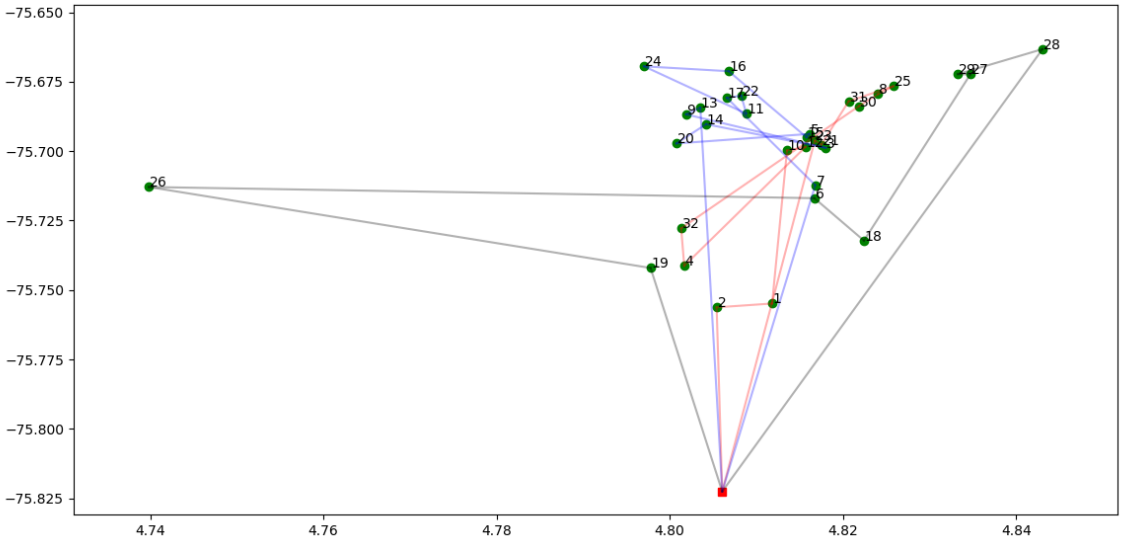

Figura 10. Gráfico datos prueba dos.

PRUEBA TRES (800 ITERACIONES)

Tabla 4. Datos prueba tres.

800 iteraciones	
Vehículos	3
Mejor Ruta	[[2, 1, 4, 24, 16, 5, 20, 14, 3, 23, 21], [7, 17, 22, 10, 11, 12, 15, 30, 8, 25, 31, 29, 27, 28], [19, 32, 26, 6, 18, 9, 13]]

Distancia Total (m)	190407
Tiempo de ejecución (Sg)	265.0577952861786
Ruta 1	
Ruta	[2, 1, 4, 24, 16, 5, 20, 14, 3, 23, 21]
Hora de Inicio (24h)	7:26:01
Hora Fin (24h)	13:18:53
Distancia (m)	58857
Ruta 2	
Ruta	[7, 17, 22, 10, 11, 12, 15, 30, 8, 25, 31, 29, 27, 28]
Hora de Inicio (24h)	6:37:41
Hora Fin (24h)	13:44:12
Distancia (m)	63757
Ruta 3	
Ruta	[19, 32, 26, 6, 18, 9, 13]
Hora de Inicio (24h)	9:07:00
Hora Fin (24h)	13:41:00
Distancia (m)	67793

Fuente: Elaboración Propia.

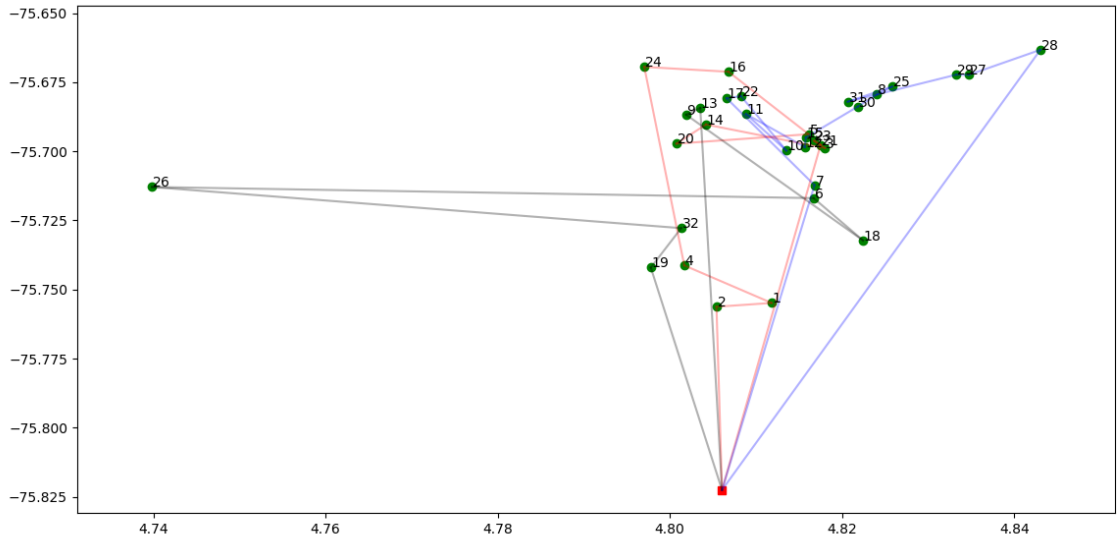

Figura 11. Gráfico datos prueba tres.

PRUEBA NUMERO 4 (1000 ITERACIONES)

Tabla 5. Datos prueba cuatro

1000 iteraciones	
Vehículos	3
Mejor Ruta	[[2, 1, 4, 24, 16, 30, 8, 25, 31, 29, 27, 28], [7, 17, 22, 10, 11, 12, 15, 5, 20, 3, 14, 13, 9, 21], [19, 26, 32, 6, 18, 23]]
Distancia Total (m)	183101
Tiempo de ejecución (Sg)	337.77412390708923
Ruta 1	
Ruta	[2, 1, 4, 24, 16, 30, 8, 25, 31, 29, 27, 28]
Hora de Inicio (24h)	7:24:37
Hora Fin (24h)	13:45:13
Distancia (m)	66131
Ruta 2	
Ruta	[7, 17, 22, 10, 11, 12, 15, 5, 20, 3, 14, 13, 9, 21]
Hora de Inicio (24h)	6:37:41
Hora Fin (24h)	13:41:27
Distancia (m)	57701
Ruta 3	
Ruta	[19, 26, 32, 6, 18, 23]
Hora de Inicio (24h)	8:54:38
Hora Fin (24h)	12:42:24
Distancia (m)	59263

Fuente: Elaboración Propia

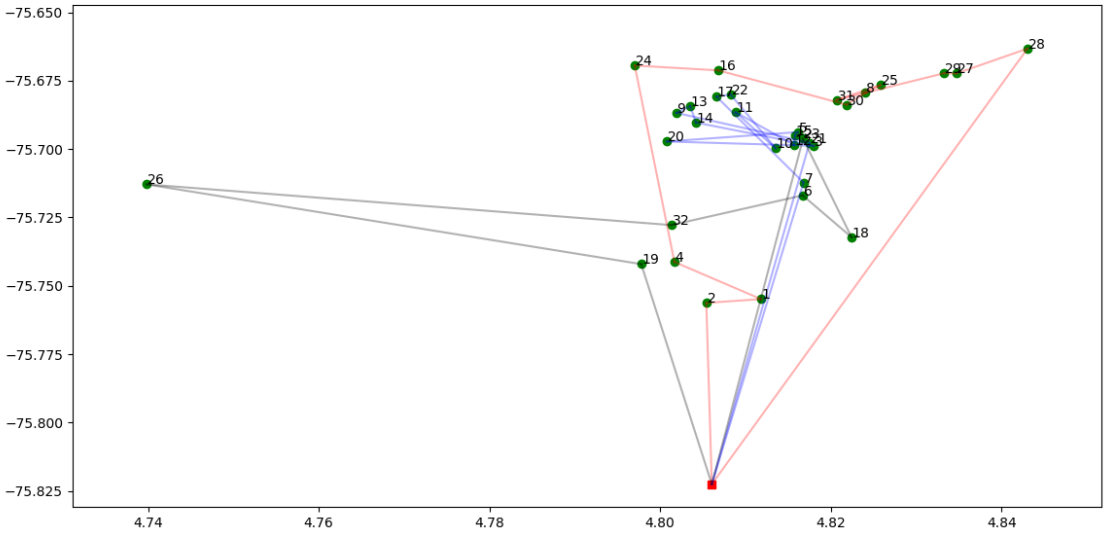

Figura 12. Gráfico prueba número cuatro.

ANALISIS DE CADA UNA DE LAS ITERACIONES

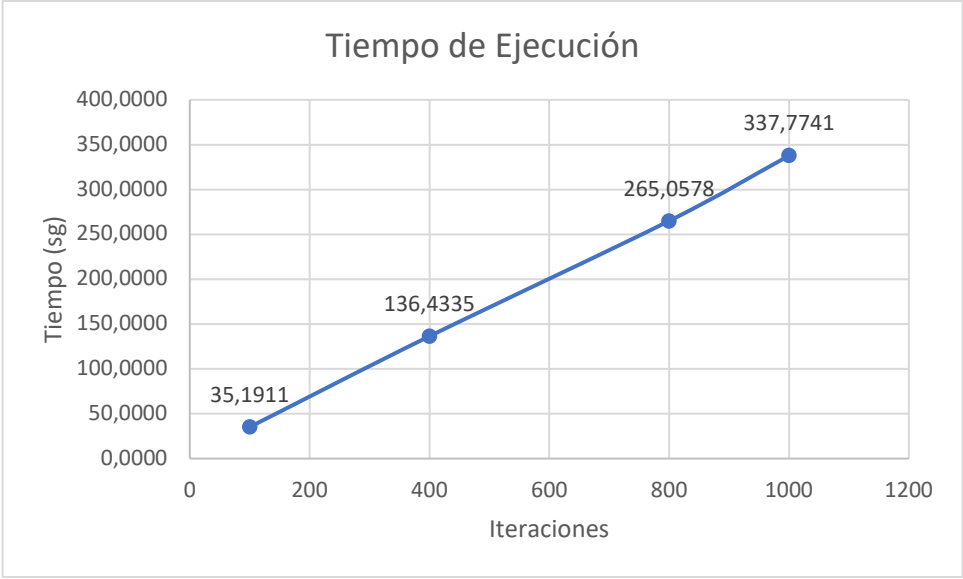

Figura 13. Tiempos de ejecución de cada iteración

Fuente: Elaboración Propia

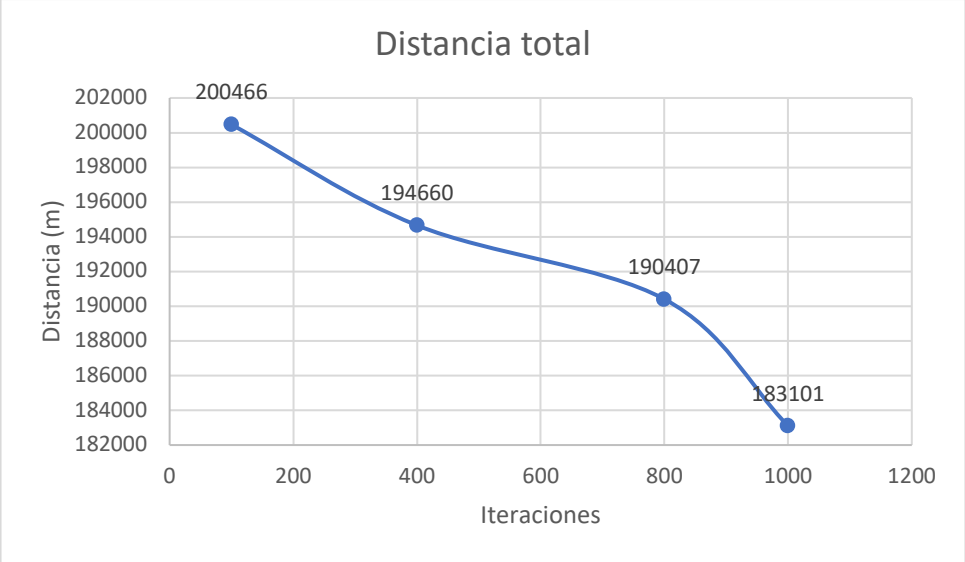

Figura 14. Distancia total recorrida por cada iteración

Fuente: Elaboración Propia

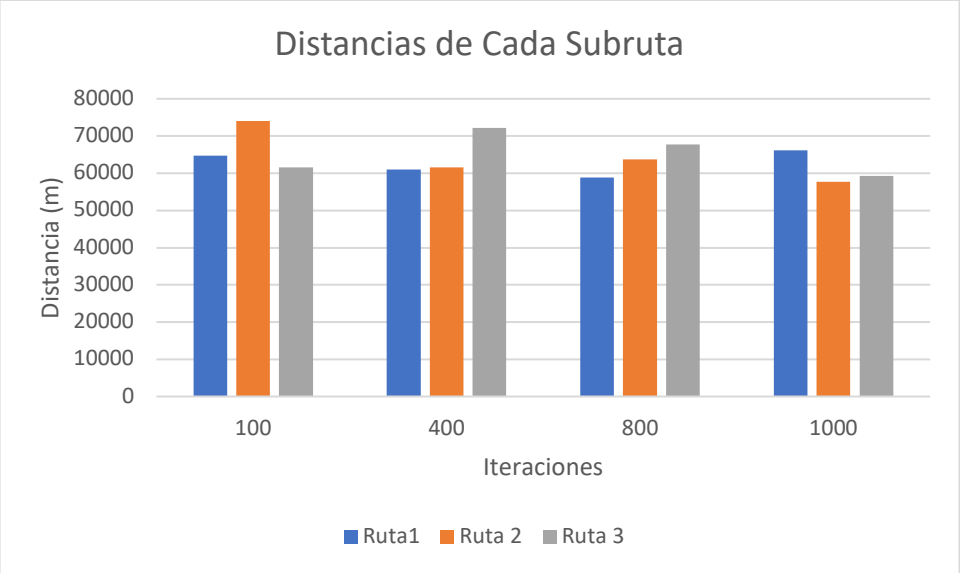

Figura 15. Distancia de cada Subruta por iteración

Fuente: Elaboración Propia

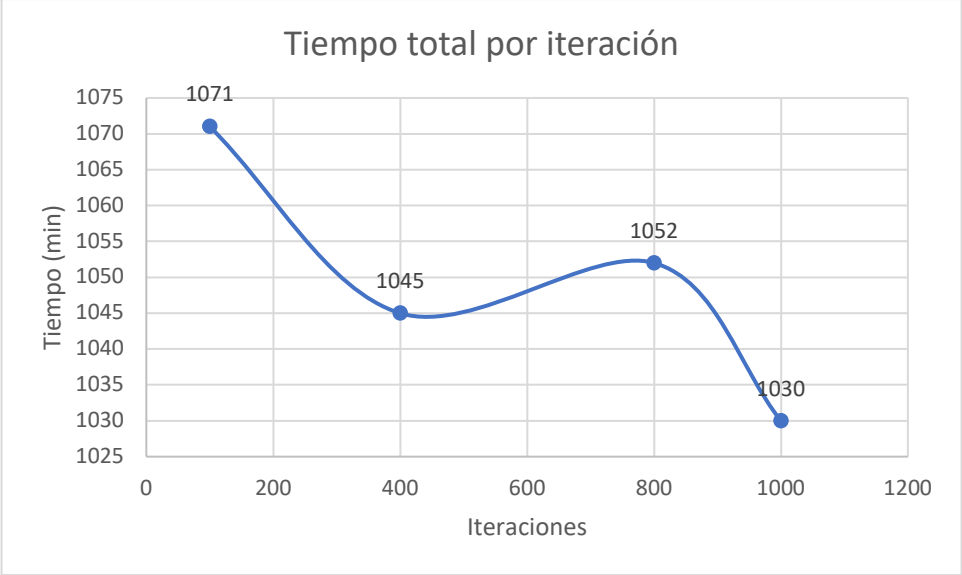

Figura 16. Tiempo total por iteración

Fuente: Elaboración Propia

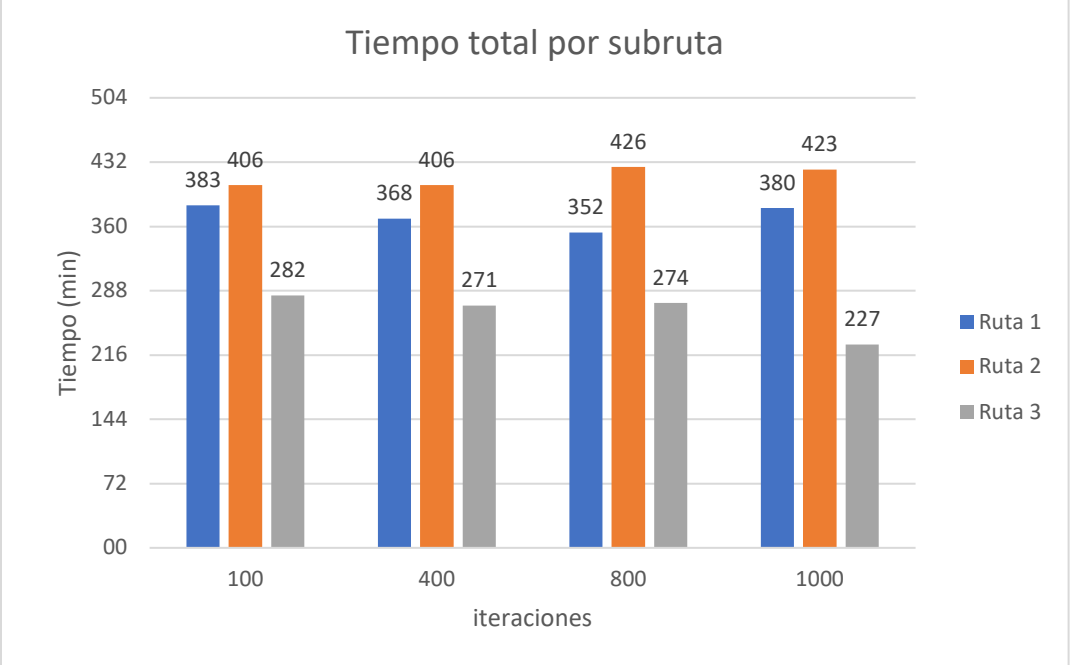

Figura 17. Tiempo Total por Subruta de cada iteración

Fuente: Elaboración Propia

9. CONCLUSIONES

- Como resultado del modelo hibrido entre Colonia de Hormigas y el Algoritmo Genético modificado de Chu-Beasley se logró una reducción de 4 a 3 vehículos y una reducción del 6 % en todo el recorrido total de recolección (de 195km a 183,1 km).
- Para los tiempos de recorrido se evidenció un comportamiento similar entre cada una de las pruebas, donde generalmente la ruta 2 (**Ver figura 17**) es la que más se tarda en completar el recorrido, analizando los puntos de recolección se puede denotar que la mayoría están ubicados en el área centro de la ciudad, además de que se asignan muchos nodos debido a la cercanía que estos tienen (todos estos son asignados por lo general a la ruta 2) **Ver Figura 12**; adicional a esto cada punto o nodo tiene un tiempo de servicio lo cual aumenta el tiempo de la ruta; Para el caso de la ruta 3 (**Ver Figura 17**) es la que menos tiempo tarda en realizar su ruta pero se contrarresta con la distancia total que debe recorrer para cumplir con los servicios de recolección (**Ver Figura 15**), se puede notar que el punto más lejano (Puesto de Salud de Altagracia, **Ver Figura 12, Nodo 26**) siempre es asignado a esta ruta y debido a su lejanía hace que el algoritmo calcule una distancia considerablemente larga tanto de ida como de regreso.
- Para las distancias de cada uno de los recorridos se evidenció que a medida que las hormigas iban aprendiendo mediante la actualización local y global de feromonas y la información de las distancias y tiempos de viaje como de servicio entre los nodos, se obtienen las mejores rutas (rutas más cortas) que,

luego, el algoritmo genético modificado de Chu-Beasley mediante los diversas etapas de selección, cruce y mutación , logra mejorar a los individuos de la población garantizando que van a estar los mejores de los mejores que se obtuvieron en colonia de hormigas, ya que como no reemplaza a todos los individuos sino al peor de ellos, reemplazándolo por el encontrado en el algoritmo puede ir comparando las mejores funciones objetivos. Gracias a esto, se pudo realizar una reducción de más de 17 km (**Ver figura 14)** y 41 minutos de recorrido (**Ver figura 16**) entre a primera iteración (100) y la cuarta iteración (1000) en donde ya el modelo hibrido comienza a converger; esto tiene como beneficio una reducción en los costos de gasolina y el tiempo de recolección entre cada uno de los clientes de aproximadamente 6%.

- Con los resultados obtenidos se puede evidenciar que la resolución de caminos fue efectiva, reduciendo la distancia total recorrida a medida que la red se entrenaba, además de que garantiza que siempre se encuentre una ruta para cada uno de los puntos y se pueda cumplir con las ventanas de tiempo. Los tiempos de ejecución tuvieron un comportamiento lineal creciente a medida que aumentaban las iteraciones, el algoritmo empieza a converger pasadas las 1000 iteraciones, donde la función objetivo empieza a tener similitudes entre una iteración y otra.
- La implementación del problema de enrutamiento de vehículos VRP puede ser aplicado a muchas áreas dentro de las que se encuentra distribución y logística empresarial, la investigación de operaciones, problemas de asignación de sistemas distribuidos, etc.; gracias a la simplicidad de la codificación de los

datos y la generación de resultados altamente eficientes, proporcionando una herramienta útil a la hora de crear soluciones optimas a problemas de optimización combinatoria.

10. BIBLIOGRAFIA

Arias, C. (2015). *Análisis e implementación del algoritmo genético de Chu-Beasley para resolver el problema del agente viajero (tsp) y su variante, el problema de rutas de vehículo (VRP)*

Orrego J., Ospina D. & Toro E. (2016) *Solución al problema de ruteo de vehículos con capacidad limitada (CVRP) usando una técnica metaheurística.* Scientia et Technica, Vol 2, No. 3. ISSN 0122-1701

Amir, W., & Azman, A. & Sahal, S. & Ngie, H. (2018). *Solving Vehicle Routing Problem using Ant Colony Optimisation (ACO) Algorithm.* Obtenido de https://www.researchgate.net/publication/328743319_Solving_Vehicle_Routing_Problem_using_Ant_Colony_Optimisation_ACO_Algorithm

Corne, D. & Dorigo, M. & Glover, F. (1999). *A Multiple AntColony System for Vehicle Routing Problems with Time Windows*. New Ideas In Optimization, McGraw-Hill, London, UK, 63-77.

Darwin, C. (1859). *On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life* (1ª Edición), Londres: John Murray.

Montes, E. (2017). *Metaheurísticas para el problema de ruteo de vehículos con ventanas de tiempo (VRP-TW)*. Obtenido de https://pdfs.semanticscholar.org/a5d9/0bc6a8080f6118594bcb2a1acf24eb43a14d.pdf

Nasser, A. & El-Sherbeny. (2010). Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. Journal of King Saud University – Science, Volume 22, 123 – 131.

Soto M., J.A, Solarte M., G.R. & Muñoz G., L.E. (2018). *Localización del punto óptimo de partida en el problema de ruteo vehicular con capacidad restringida (CVRP).* Tecnura, 23(59), 27-46.

Solarte Martínez, G. R., Castillo Sanz, A. G., & Rodríguez Gahona, G. (2015). *Optimization of a vehicular routing using simple genetic chu-beasley algorithm*. Revista Tecnura, 19 (44), 93-108. doi:http://dx.doi.org/10.14483/udistrital.jour.tecnura.2015.2.a07

Gambardella L., Taillard R., & Agazzi G. (2001). *MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows*. Obtenido de https://www.researchgate.net/publication/2369336_MACS-VRPTW_A_multiple_ant_colony_system_for_vehicle_routing_problems_with_time_windows

Guasmayan F. (2014). *Solución del problema de ruteo de vehículos dependientes del tiempo utilizando un algoritmo genético,* Obtenido de:

http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/4562/5196G917.pdf?sequence=1&isAllowed=y

Keskin, M. & Çatay, B. (2016). *Partial recharge strategies for the electric vehicle routing problem with time windows,* Transportation Research Part C 65, 111 – 127.

Peña A., I.G., Gutiérrez S., A., Lopez S., C.A., & Rocha M., L.B. (2017). *CVRPTW model applied to the collection of food donations.* Obtenido de http://ieomsociety.org/bogota2017/papers/271.pdf

Mohapatra, G. S. (2014*). Models for Practical Routing Problems in Logistics*.

Barán, B., & Almirón, M. (2001). *Colonias distribuidas de hormigas en un entorno paralelo asíncrono.* Obtenido de https://www.cnc.una.py/publicaciones/1_46.pdf

Dantzig G. B. y Ramser J. H. (1959). The truck Dispatching Problem. (Ed. 6)

Rajeev G., Raman M. (2018) *A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems.* Obtenido de https://www.sciencedirect.com/science/article/abs/pii/S187775031730813X

Soto D., Soto W. & Pinzón Y. (2008) *Una metaheurística aplicada a un problema de planificación de rutas.* Obtenido de bdigital.unal.edu.co/15493/1/10107-18480-1-PB.pdf

Castiblanco A., Martin D., García J. (2017) *Ruteo de vehículos en el sector de hidrocarburos aplicando colonia de hormigas.* Obtenido de

https://repository.javeriana.edu.co/bitstream/handle/10554/36495/CastiblancoSuarezAndresFelipe2017.pdf?sequence=1&isAllowed=y

Ruiz, A., Toro, M. & Salazar, H. (2007). *Algoritmo genético modificado Chu-Beasley aplicado a la identificación de errores en la estimación de estado en sistemas*. Obtenido de https://revistas.utp.edu.co/index.php/revistaciencia/article/view/5343/2959

Atehortua, A. (2012). *Optimización Basada en Colonia de Hormigas: Generalidades y estudio del algoritmo sistema hormiga y aplicación a un job shop.* Obtenido de: https://www.academia.edu/6824195/OPTIMIZACI%C3%93N_BASADA_EN_COLONIA_DE_HORMIGAS_GENERALIDADES_Y_ESTUDIO_DEL_ALGORITMO_SISTEMA_HORMIGA_Y_APLICACI%C3%93N_A_UN_JOB_SHOP_ANT_COLONY_OPTIMIZATION_GENERALITIES_AND_STUDY_OF_ANT_SYSTEM_ALGORITHM_AND_A_JOB_SHOP_APLICATION_ANDR%C3%89S_ATEHORT%C3%9AA