

**DESARROLLO DE UN ALGORITMO MEMÉTICO PARA SOLUCIONAR EL
PROBLEMA DE RUTEO DE VEHÍCULOS CON DEMANDAS
ESTOCÁSTICAS Y MÚLTIPLES COMPARTIMIENTOS**

KONY VANESSA GARCÍA RODRÍGUEZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO - MECÁNICAS
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES
BUCARAMANGA
2015**

**DESARROLLO DE UN ALGORITMO MEMÉTICO PARA SOLUCIONAR EL
PROBLEMA DE RUTEO DE VEHÍCULOS CON DEMANDAS ESTOCÁSTICAS Y
MÚLTIPLES COMPARTIMIENTOS**

KONY VANESSA GARCÍA RODRÍGUEZ

**Trabajo de Grado para optar al título
de Ingeniero Industrial**

**Director
Ph.D. HENRY LAMOS DÍAZ**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO - MECÁNICAS
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES
BUCARAMANGA
2015**

DEDICATORIA

A Dios por estar conmigo en cada paso que di, porque escucho cada una de las suplicas y nunca me dejó rendir.

A mi madre y mi padre por ser mi compañía y mi inspiración para cumplir mis metas.

A mi hermano porque es un motivo más por el cual voy a seguir creciendo.

A mis abuelos que son mis padres y que diariamente me brindan su amor incondicional.

A mi bisabuela que es un ángel, que me acompaño siempre.

A mis tíos, primos y demás familiares porque siempre conté con su cariño.

A mis amigos que me enseñaron y me brindaron su valiosa amistad.

AGRADECIMIENTOS

A Dios por permitir que este proyecto sea una realidad cumplida.

Al Docente Henry Lamos por guiarme en el camino recorrido.

Al Ingeniero Daniel Buitrago por su asesoría.

A la Universidad Industrial de Santander, la Escuela de Estudios Industriales y Empresariales y al grupo de investigación Ópalo por brindarme los conocimientos y la formación académica pertinente, para lograr mis metas.

CONTENIDO

	PÀG.
INTRODUCCIÓN	14
1. GENERALIDADES DEL PROYECTO	16
1.1 PLANTEAMIENTO DEL PROBLEMA	16
1.2 JUSTIFICACIÓN DEL PROYECTO	16
1.3 OBJETIVOS	17
1.3.1 Objetivo General	17
1.3.2 Objetivos Específicos	17
1.4 METODOLOGÍA	17
1.4.1 Revisión Bibliográfica	17
1.4.2 Documentación	18
1.4.3 Formulación	18
1.4.4 Programación	18
1.4.5 Experimentos Computacionales	18
2. MARCO TEÓRICO	19
2.1 OPTIMIZACIÓN COMBINATORIA	19
2.1.2 Problemas Combinatorios	19
2.1.3 Complejidad Computacional	20
2.1.4 Métodos de Solución y/o Técnicas de Optimización	21
2.2 SISTEMAS LOGÍSTICOS	25
2.3 DISTRIBUCIÓN FÍSICA	27
2.3.1 Medios De Transporte	27
2.3.2 Elección del Medio de Transporte	29
2.3.3 Canales de Distribución	30
2.4 PROBLEMA DE RUTEO DE VEHÍCULOS	31
2.4.1 Definición de un problema de ruteo	32
2.4.2 Variantes del VRP	33
2.4.2 Modelación matemática del VRP	34

2.5 PROBLEMA DE RUTEO DE VEHÍCULOS CON DEMANDAS ESTOCÁSTICAS (VRPSD) Y MC-VRPSD	36
2.6 ALGORITMOS MEMÉTICOS [MA]	38
2.7 ALGORITMO DE OPTIMIZACIÓN DE ENJAMBRE DE PARTÍCULAS (PSO)	39
2.8 ALGORITMO DE OPTIMIZACIÓN DE ENJAMBRE DE PARTÍCULAS UNIFICADO (UPSO).....	41
3. DISEÑO DE LA SOLUCIÓN.....	43
3.1 DESARROLLO DEL ALGORITMO COMPUTACIONAL	52
4. RESULTADOS COMPUTACIONALES	53
5. CONCLUSIONES.....	59
BIBLIOGRAFIA	60

LISTA DE TABLA

	Pág.
Tabla 1 Datos de entrada UPSO	43
Tabla 2 Datos ejemplo	53
Tabla 3 Datos iniciales	54
Tabla 4 Rutas asignadas	54
Tabla 5 Demandas de las rutas asignadas	55
Tabla 6 Matriz de distancias	56
Tabla 7 Distancias de las rutas	57

LISTA DE ILUSTRACIONES

	Pág.
Ilustración 1 Técnicas de Optimización.....	22
Ilustración 2 Procedimiento de un método exacto	22
Ilustración 3 Esquema de la Cadena de Suministro	26
Ilustración 4 Medios de Transporte.....	28
Ilustración 5 Representación Gráfica del VRP	33
Ilustración 6 interface del algoritmo computacional creado	52
Ilustración 7 interface de ejemplo	58

TABLA DE CUMPLIMIENTO DE OBJETIVOS

OBJETIVO	DESCRIPCION	CUMPLIMIENTO
1	Construir el estado del arte actual sobre la aplicación de los algoritmos meméticos para la solución del problema combinatorios	CAP 2
2	Elaborar el modelo de solución para el problema estocástico de ruteo de vehículos con múltiples compartimentos en base a un algoritmo memético	CAP 3, ANEXO B
3	Construir un banco de instancias para la validación del modelo.	ANEXO A
4	Desarrollar un modelo computacional para la solución del problema de ruteo de vehículos con múltiples compartimentos en Matlab	CAP 3.1, ANEXO B

RESUMEN

TITULO:

DESARROLLO DE UN ALGORITMO MEMÉTICO PARA SOLUCIONAR EL PROBLEMA DE RUTEO DE VEHÍCULOS CON DEMANDAS ESTOCÁSTICAS Y MÚLTIPLES COMPARTIMIENTOS.*

AUTOR:

KONY VANESSA GARCIA RODRIGUEZ.**

PALABRAS CLAVES:

ALGORITMO MEMETICO, RUTEO DE VEHICULOS DE MULTIPLES COMPARTIMIENTOS, DEMANDAS ESTOCÁSTICAS.

DESCRIPCION:

En este documento se detalla el desarrollo de un programa estocástico con recurso y se resuelve mediante un algoritmo memético (MA), codificado en Matlab. Este método permite desarrollar la búsqueda de un conjunto de rutas con el costo mínimo para satisfacer la demanda que existe sobre varios tipos de productos para una red de clientes.

En este trabajo se extiende el MC-VRP mediante la introducción de la incertidumbre sobre lo que se conoce como el MC-VRP con demandas estocásticas (MC-VRPSD) el cual consiste en el diseño de las rutas de transporte para satisfacer las demandas de un grupo de clientes para varios productos que, debido a la incompatibilidad de las limitaciones, se deben cargar en compartimentos de vehículos independientes.

El MC-VRPSD (problema de ruteo de vehículos de múltiples compartimentos con demandas estocásticas) es considerado un problema optimización combinatoria clasificado de tipo NP-HARD por lo cual desde el punto de vista computacional se puede llegar a obtener una solución factible en un tiempo razonable, con la ayuda de los algoritmos meméticos, los cuales intensifican la búsqueda de la mejor solución demostrando su efectividad y eficiencia en problemas de gran magnitud.

En la actualidad las empresas de logística no solo buscan programas computaciones que generen la rutas que deben seguir sus vehículos, sino que también esos modelos se asemejen a la realidad con el fin de que los recursos con los que cuenta la compañía y los productos transportados sean tenidos en cuenta para ofrecer al cliente una experiencia satisfactoria en términos de calidad de entrega, tiempo y así mismo disminución en sus costos operativos.

* Proyecto de Grado

** Facultad de ingeniería Físico Mecánicas, Escuela de Estudios Industriales y Empresariales. Director: Henry Lamos Díaz

ABSTRACT

TITLE:

DEVELOPMENT OF A MEMETIC ALGORITHM TO SOLVE FOR THE MULTI-COMPARTMENT VEHICLE ROUTING PROBLEM WITH STOCHASTIC DEMANDS*

AUTHOR:

KONY VANESSA GARCIA RODRIGUEZ.**

KEYWORDS:

MEMETIC ALGORITHM, MULTI-COMPARTMENT VEHICLE ROUTING PROBLEM, STOCHASTIC DEMANDS

DESCRIPTION:

This document details the development of a stochastic program with recourse and is solved by a memetic algorithm (MA), coded in Matlab. This method allows search to develop a set of routes with minimum cost to meet the demand that exists on various kinds of products to a network of customers.

This work extends the MC-VRP by introducing uncertainty on what it is known as the MC-VRP with stochastic demands (MC-VRPSD) which consists of designing transportation routes to satisfy the demands of a set of customers for several products that, because of incompatibility constraints, must be loaded in independent vehicle compartments.

The MC-VRPSD (The multi-compartment vehicle routing problem with stochastic demands) is considered an optimization problem combinatorial classified type NP-HARD so from a computational point of view can earn a feasible solution in a reasonable time with the help of the memetic algorithms, which intensify search the best solution to demonstrate its effectiveness and efficiency in large-scale problems

At present the companies of logistics not only look for programs computations that generate the routes to be followed by their vehicles, but also these models should be alike the reality in order which the resources with which it counts the company and the transported products are had in account to offer a satisfactory experience to the client in quality terms of delivery, time and likewise diminishment in their operating costs.

* Grade Project

** Facultad de ingeniería Físico Mecánicas, Escuela de Estudios Industriales y Empresariales. Director: Henry Lamos Díaz

INTRODUCCIÓN

En la actualidad el enfoque de las empresas de distribución y logística ha cambiado debido a la competencia que se ha venido presentando en este tipo de mercado, y también a los costos externos e internos con los que se debe trabajar para proporcionar un buen servicio. Al momento de planear y realizar el trabajo de distribución de la mercancía se debe tener presente no solo los medios de transporte con los que la empresa cuenta, sino también los tipos de productos y/o materias primas que interfieren en la cadena de suministro.

Los operadores logísticos buscan no solo reducir al máximo sus gastos sino también, brindarles a sus clientes una experiencia satisfactoria marcada por la calidad y el cumplimiento en las entregas, es por esto que el transporte y las rutas juegan el papel más importante dentro de la logística de distribución, motivo por el cual muchas de estas empresas han optado por modelos computacionales que brindan una solución al problema de ruteo vehicular que en muchos casos no tiene en cuenta el tipo de carga y la demanda generada por los productos.

El país se encuentra atrasado en cuanto al desempeño logístico. El reto para las empresas y el país en general será mejorar la infraestructura y fortalecer las actividades relacionadas al transporte y comercio internacional. Colombia no es ajena a esta situación, estando en una búsqueda constante de una mayor competencia en el tema logístico. De ahí que uno de los avances más importantes del país es la elaboración del documento Conpes 3547¹, denominado “Plan Nacional de Logística”, aprobado en 2008, y que definió los lineamientos de política, las necesidades en materia de infraestructura y el financiamiento para el desarrollo de acciones que permitieran aumentar la competitividad mediante la adopción de mejores prácticas de transporte y logística.

Dado que vivimos en un mundo globalizado, que se encuentra en constante cambio, la cadena de suministro es la pieza clave para hacer que las empresas sean cada día más competitivas, debido a esto se ha cambiado la perspectiva que se tiene de la cadena de suministro enfocándose en cuatro aspectos claves: 1) Los ciclos y tiempos de entrega entre proveedor y cliente son cada vez más cortos; 2) los vínculos comerciales entre proveedor y cliente son cada vez más estrechos; 3) el comercio electrónico crece a pasos agigantados y 4) la integración de los sistemas productivos pasó de ser regional a volverse global. Estos factores explican la importancia que se le da en las empresas a la logística de la cadena de

¹ Departamento Nacional de Planeación [en Línea] [citado el 30 de noviembre de 2013] Disponible en internet: <https://www.dnp.gov.co/Portals/0/archivos/documentos/Subdireccion/Conpes/3547.pdf>

abastecimiento, y ejemplo de ello es que las compañías a nivel mundial han pasado de considerarla un área más de la organización, a verla como la “gerencia de la cadena de abastecimiento”².

Por tal circunstancia este proyecto se basa en generar una solución óptima con un mínimo de tiempo computacional mediante el programa Matlab fundamentado en algoritmos meméticos que son la base para desarrollar rutas de servicios, teniendo en cuenta tanto las restricciones de capacidad debido a la incompatibilidad de productos, como a las demandas de cada cliente, debido a la complejidad de este problema, la utilización de algoritmos meméticos se sustenta en que son más eficientes al momento de trabajar con problemas no lineales y de gran tamaño que los algoritmos evolutivos tradicionales.

² Revista de logística [en línea] edición 21. LEGIS. Julio 2013. [citado: 27 nov. 2013]. Disponible en internet: <http://www.revistadelogistica.com/La-logistica-de-la-cadena-de-abastecimiento-un-desafio-gerencial-del-siglo-XXI.asp>

1. GENERALIDADES DEL PROYECTO

1.1 PLANTEAMIENTO DEL PROBLEMA

Plantear el diseño de rutas de servicio (recolección-entrega) para ofrecer a un conjunto de clientes que poseen demandas estocásticas independientes para diferentes tipos de productos que deben ser transportados en compartimentos independientes debido a las restricciones de incompatibilidad de productos.

1.2 JUSTIFICACIÓN DEL PROYECTO

Cada año se mueven miles de toneladas de carga, lo que involucra una organización impecable para hacer posible el cumplimiento de los servicios que ofrecen las empresas; muchos elementos se deberán tener en cuenta para lograr este propósito. Un elemento importante en la cadena logística es la red de transporte, en lo que le concierne al presente trabajo se encuentra el ruteo de vehículos. Se han desarrollado modelos que capturan diferentes situaciones del mundo real, por ejemplo, el modelo para elaboración de rutas con vehículos de capacidad finita, con ventanas de tiempo, con demanda estocástica de los clientes, con recolección y entrega de mercancía, con flotas heterogéneas, con varios compartimentos, etcétera.

En el presente trabajo se propone diseñar en base a algoritmos meméticos rutas de servicio para clientes con demandas estocásticas independientes, diferentes tipos de productos que deben ser transportados en compartimentos independientes debido a restricciones de incompatibilidad.

La importancia del presente trabajo es su alta complejidad tanto teórica como computacional lo que hace que sea muy atractivo empezar a desarrollar la presente línea en el grupo de investigación.

El problema VRPSD (por sus siglas en inglés Vehicle Routing Problem with Stochastic Demands o Ruteo de Vehículos con Demandas Estocásticas) es un problema tipo combinatorio y se clasifica en la categoría de problemas NP-hard, esto es son difíciles de resolver por medio de métodos exactos debido a su gran complejidad computacional. En lugar de los métodos exactos se usan algoritmos

evolutivos conocidos como algoritmos meméticos que permiten obtener una solución factible “buena” en un tiempo razonable de cómputo.

Los algoritmos meméticos han sido aplicados en diferentes áreas y han demostrado ser más eficientes y efectivos que los algoritmos evolutivos tradicionales, debido a que son capaces de trabajar con problemas de gran tamaño y no lineales.

1.3 OBJETIVOS

1.3.1 Objetivo General

Desarrollar un algoritmo memético para la solución del problema de ruteo de vehículos de múltiples compartimientos con demandas estocásticas.

1.3.2 Objetivos Específicos

- Construir el estado del arte actual sobre la aplicación de los algoritmos meméticos para la solución de problemas combinatorios.
- Elaborar el modelo de solución para el problema estocástico de ruteo de vehículos con múltiples compartimientos en base a un algoritmo memético
- Construir un banco de instancias para la validación del modelo.
- Desarrollar un modelo computacional para la solución del problema de ruteo de vehículos con múltiples compartimientos en Matlab.

1.4 METODOLOGÍA

Durante la elaboración de este proyecto de grado se tiene en cuenta la siguiente metodología.

1.4.1 Revisión Bibliográfica

En la revisión de la bibliografía se descubrió que no ha recibido mucha atención en la literatura por parte de los autores, sin embargo se efectúan estudios por cada factor que comprende el problema de ruteo de vehículos de varios compartimientos basado en algoritmos meméticos con demandas estocásticas y, se descubren otras investigaciones de este tipo con las cuales se documentaran todos los aportes encontrados.

1.4.2 Documentación

Se recopila toda la información obtenida en la profundización bibliográfica, en un documento donde se explica de manera clara y precisa la función del algoritmo memético para el problema de MC-VRPSD.

1.4.3 Formulación

Definir las características y variables del modelo matemático, así como formular el objetivo y las restricciones para programar el algoritmo memético que se va a desarrollar.

1.4.4 Programación

Desarrollo de las características del algoritmo en el lenguaje de MATLAB.

1.4.5 Experimentos Computacionales

Se realizan varias corridas del algoritmo con el fin de generar el banco de instancias y así probar que el código no presente errores.

2. MARCO TEÓRICO

2.1 OPTIMIZACIÓN COMBINATORIA

La optimización combinatoria es un método de la matemática moderna aplicada y de la cultura computacional, que trabaja en problemas que se identifican por tener un conjunto finito de soluciones factibles.

La optimización combinatoria es una rama de la optimización cuyos problemas se centran en problemas de decisión, problemas tales como la minimización en el uso de recursos empresariales, la ruta más corta en la distribución de productos y/o servicios, el Planning y Scheduling en empresas y el ruteo de vehículos en compañías de transporte hacen parte de este tipo de problemas. Los problemas se tornan más difíciles cuando se tiene en mente, como ocurre a nivel real, la optimización de más de un objetivo a la vez, lo cual convierte el problema combinatorio a problema combinatorio de optimización multi-objetivo³.

Un problema de optimización combinatoria es uni-objetivo cuando sobre el espacio de distribuciones se construye una sola función de valor; y muti-objetivo cuando es necesario más de una. Los problemas de optimización combinatoria están ligados con la asignación de recursos para cumplir el objetivo deseado y con la particularidad de que los valores de algunas de las variables son restringidos a ser enteros. Las restricciones en los recursos disminuyen el número de opciones consideradas como factibles, y la principal meta a determinar es cuál de las alternativas es la mejor. Aunque por la grandeza de este tipo de problemas en teoría, estos no ofrecen dificultad para su resolución, en la práctica, suele faltar tiempo para encontrar la respuesta; por ello se necesitan otras técnicas para resolver estos problemas de tipo computacional.

2.1.2 Problemas Combinatorios

Los problemas combinatorios tratan de encontrar un objeto entre un conjunto finito de posibilidades. Este conjunto finito se define como la delimitación del problema de optimización. Los objetos pueden ser números naturales, estructuras de grafos o

³ NIÑO RUIZ ,Elías David, Optimización Combinatoria: una perspectiva desde la teoría de autómatas, Editorial Académica Española, Publicado en: Enero 22 de 2012, P 228.[en Línea] [Citado: 1 de diciembre de 2013] Disponible en: http://www.faae.org.co/formularios/adjuntos_fciencias/512/EliasNinoResumen.pdf

permutaciones, por lo tanto, debido a su naturaleza discreta llegan a ser muy complejos para encontrar la solución.

Los problemas de tipo combinatorios cuentan con los siguientes elementos:

- a. Un conjunto de variables independientes y/o de decisión
- b. Una función objetivo que mide la efectividad de cada sistema de decisiones.
- c. Un conjunto de restricciones que representan las limitaciones bien sea de capital, recursos, etc.

En general los problemas de optimización combinatoria son clasificados de acuerdo a su complejidad computacional, y esto ha llevado al desarrollo de muchos algoritmos para hallar la solución óptima del problema. La optimización combinatoria no solo es útil para comprender la complejidad de los algoritmos, sino también permite verificar si una propuesta de solución de un problema de optimización discreta es óptima⁴.

2.1.3 Complejidad Computacional

La complejidad computacional de un algoritmo se da debido a la dificultad del problema a tratar, las variables y las restricciones con las que cuenta este. En la teoría se pueden dar solución a varios tipos de problemas, mientras que en la práctica son intratables debido a su composición.

Es por esto que se clasifica de la siguiente manera⁵:

- a. P (Polinomial- time): esta clase corresponde al tipo de problemas cuya complejidad permite darle una solución en un tiempo polinomial, es decir puede ser resuelto en un tiempo prudente por medio de los computadores que existen hoy en día⁶.
- b. NP (Non-Deterministic Polinomial- time): este corresponde a un problema para el cual las soluciones son corroboradas por medio de un algoritmo en un tiempo de proceso polinomial, lo anterior no implica que las soluciones

⁴ LANGE, Kenneth. Applied Probability: Combinatorial Optimization. En: Springer Science Business Media: Cap. 5 (2010), p. 103.122.

⁵ GOMEZ ATUESTA, David Fernando; RANGEL CARVAJAL, Carlos Eduardo. Formular las Mateheurísticas Búsqueda Tabú y Reconocido Simulado para la solución del CVRP. Tesis de pregrado de Ingeniería Industrial. Bucaramanga. Universidad Industrial de Santander. 2011

⁶ Departamento de Computación. Facultad de Ciencias Exactas y Naturales - Universidad de Buenos Aires. [en línea] [citado: Nov. 19, 2013] http://www.dc.uba.ar/materias/aed3/2013/1c/teorica/algo3_compl.pdf

serán encontradas rápidamente, solo que la solución hallada puede ser verificada o rechazada de forma rápida. Un problema de esta clase presenta un tiempo de proceso exponencial en un computador personal y generalmente no se soluciona en un tiempo razonable, pero si pueden ser resueltos en una máquina de Turing.

- c. NP- Complete (Non-Deterministic Polinomial- time complete): en este tipo de problemas de relación entre el número de parámetros y la complejidad del problema es exponencial. Si alguna búsqueda iterativa es adoptada, entonces el incremento exponencial en la complejidad del problema trae como resultado un incremento exponencial en el tiempo de solución.
- d. NP-Hard (NP-difíciles): Los problemas de tipo NP-hard son problemas en parte similares, pero más difíciles que los problemas NP-Complete. Muy a menudo, los problemas NP-hard realmente requieren un tiempo exponencial o incluso uno mayor. Teniendo en cuenta que los problemas NP-Complete son un subconjunto de los problemas NP-hard, y es por eso que los problemas NP-Complete son a veces llamados NP-hard⁷.

Los problemas de distribución logística, de ruteo, o de encaminamiento, como el problema del agente viajero (TSP), el problema de vehículos (VRP) o el problema de ruteo de vehículos de múltiples compartimientos (MC-VRP), son problemas de orden NP-Hard debido a la dificultad que se tiene de enumerar todas las configuraciones de las soluciones, debido a las múltiples interrelaciones de sus variables y el gran tamaño que tienen los datos de entrada.

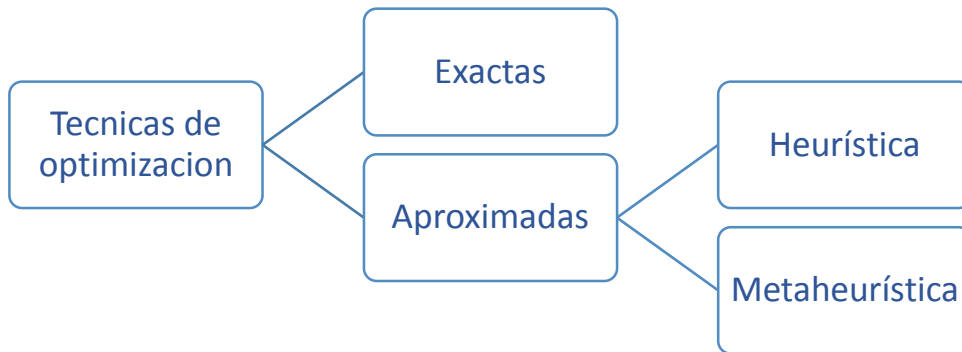
2.1.4 Métodos de Solución y/o Técnicas de Optimización

Debido a la importancia de los problemas de optimización combinatoria en aplicaciones prácticas, científicas e industriales, se han desarrollado múltiples métodos para desarrollarlos, dividiéndose en dos clases exactas y aproximadas.

Las técnicas exactas encuentran la solución óptima para cualquier instancia de un problema en un tiempo determinado, mientras que los métodos aproximados permiten encontrar una solución factible, que en algunos casos pueden considerarse cercana a la solución óptima, y es posible ser hallada en un tiempo razonable y con un moderado recurso computacional.

⁷ Class NP, NP-complete, and NP-hard problems W. Hämaläinen, Nov 2006 [en Linea] [citado: Nov. 20, 2013] <http://cs.joensuu.fi/pages/whamalai/daa/npsession.pdf>

Ilustración 1 Técnicas de Optimización

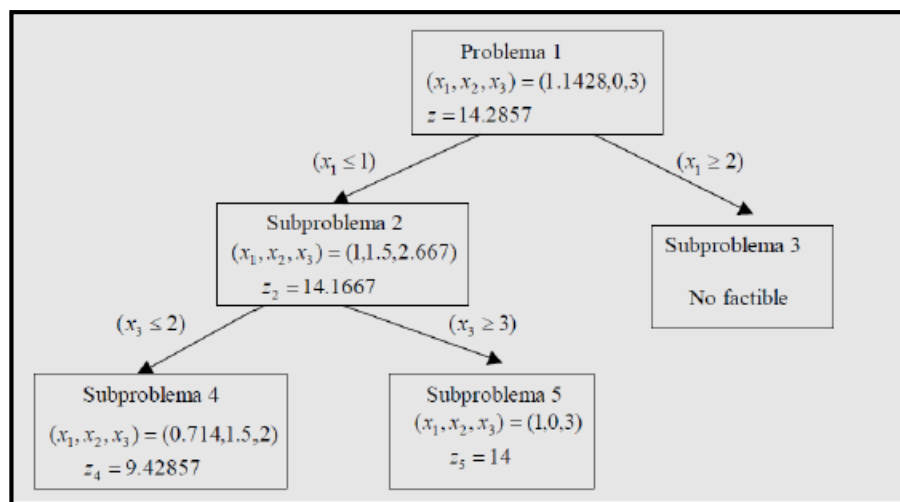


Fuente: Autor

✓ Métodos Exactos.

Los métodos exactos son técnicas de búsqueda refinadas; esto quiere decir que, dividen el espacio de búsqueda en pequeños problemas aún más sencillos, y que al solucionarlos, llevan a la solución final. Un método exacto divide el problema general en pequeños subproblemas, cada uno con restricciones y características definidas como lo muestra el siguiente ejemplo (ilustración 2).

Ilustración 2 Procedimiento de un método exacto



Fuente: GARCÍA, José Pedro. Modelos y Métodos de Investigación de Operaciones. Grupo de Investigación ROGLE. Departamento de Administración de Empresas.

✓ **Métodos Heurísticos**

Se define la heurística como una estrategia, método, criterio o truco utilizado para hacer más sencilla la solución de problemas difíciles por medio de uno o varios algoritmos que permite conseguir resultados de buena calidad para un problema dado, y de esta manera obtener menores tiempos de ejecución, pero sin asegurar la optimalidad de la respuesta.

Debido a la existencia de algunos problemas significativos con un gran interés práctico difíciles de solucionar, empiezan a nacer algoritmos capaces de ofrecer posibles respuestas que aunque no obtienen el resultado óptimo, se aproximan en un tiempo de cómputo razonable. Estos algoritmos están fundamentados en el conocimiento heurístico y por lo tanto adoptan el calificativo de algoritmos heurísticos.

Frecuentemente, los algoritmos heurísticos hallan buenas soluciones, sin embargo algunas veces no hay pruebas de que la solución alcance a encontrarse en un tiempo razonablemente corto o incluso asegurar que no sea errónea.⁸

A diario se conocen nuevas técnicas heurísticas que son utilizadas por científicos, investigadores operativos y profesionales, para resolver todo tipo de problemas que antes eran excesivamente complejos o magnos para las anteriores generaciones.

✓ **Métodos Metaheurísticos**

Una Metaheurística es una estrategia general para la construcción de algoritmos y la resolución de una gran variedad de problemas para los que no existe un algoritmo confiable, ya sea por la complejidad debido a que requieren tiempos de cálculos grandes y el uso de una gran cantidad de recurso computacional.

⁸ METODOS HEURISTICOS [en línea] [citado: Nov. 08, 2013] Disponible en internet: topicoslacruz.com/post/METODO%20HEURÍSTICO.doc

Algunas de las Metaheurísticas más utilizadas en los problemas de optimización combinatoria, son⁹:

- Algoritmos genéticos: corresponden a una clase de algoritmos evolutivos, los cuales fueron descritos por primera vez por A.S.Fraser en 1957. Cada solución del problema se codifica en un cromosoma, donde a cada elemento de éste se le llama gen. El conjunto de cromosomas forman una población, que para una iteración corresponde a una generación. Se les aplican diversos operadores, para generar nuevos individuos, que son agregados a la población, en un proceso iterativo que trata de escapar de mínimos locales. Han tenido éxito en resolver problemas de rutas de vehículos, así como de localización y cobertura, entre muchos otros. Por otra parte, se ha trabajado en ellos como algoritmos meméticos (algoritmos genéticos con algún procedimiento adicional de búsqueda local), para la extensión del problema que considera una flota heterogénea de vehículos.
- Búsqueda en vecindarios variables: comúnmente denominada VNS por su sigla en inglés, parte desde una solución inicial aleatoria, a partir de la que se van explorando, usando algún algoritmo de búsqueda local eficiente, vecindarios progresivamente más lejanos (y grandes); en caso de que se encuentre una mejor solución, la búsqueda se mueve hasta ella, reiniciándose la búsqueda en los vecindarios de esta.
- Recocido simulado: se trata de asemejar el proceso de manufactura donde un material (metal) es calentado hasta altas temperaturas, para luego ser enfriado lentamente, de tal manera que sus estructuras cristalinas se reorganicen en la configuración de mínima energía. En la versión computacional, la temperatura es discretizada, y para realizar una analogía con la situación física, se admiten soluciones peores que la mejor encontrada con una probabilidad proporcional a la distribución termodinámica de Boltzmann, permitiendo escapar de óptimos locales. Es fácil de implementar, y posee varios parámetros que se pueden cambiar para buscar mejoras, como el patrón de enfriamiento, o la probabilidad de aceptar una peor solución. Es un método de trayectoria.
- Búsqueda tabú: en esta Metaheurística, se busca en la proximidad de la solución actual otra que mejore la evaluación de la función objetivo, almacenando las soluciones anteriores (o alguna característica de éstas), las que son marcadas como

⁹El problema de rutas de vehículos: Extensiones y métodos de resolución, estado del arte [en línea]. Disponible en internet http://ceur-ws.org/Vol-558/Art_23.pdf

tabú. Esto evita que el algoritmo entre en un ciclo, pudiendo escapar de óptimos locales.

- Colonias de hormigas: basadas en la naturaleza, varias hormigas (procesos, hilos, agentes, etc.) exploran distintas direcciones del espacio de soluciones factibles, dejando tras de sí un rastro de feromonas, que le indican a la siguiente hormiga las direcciones más 'interesantes' de ser exploradas, las que toma con una probabilidad proporcional al nivel de feromona existente, en un intento por no caer en un óptimo local. Junto con esto, por su naturaleza multi-agente de búsqueda aparece como una estrategia trivialmente paralelizable. Además, los niveles de feromonas disminuyen tras cada iteración (se evapora). Como trabaja con distintos agentes, se le considera un algoritmo basado en poblaciones.

- Enjambre de partículas: se busca simular la búsqueda realizada por entes colaborativos, considerando las interacciones entre ellos y como se orientan hacia una búsqueda eficiente.

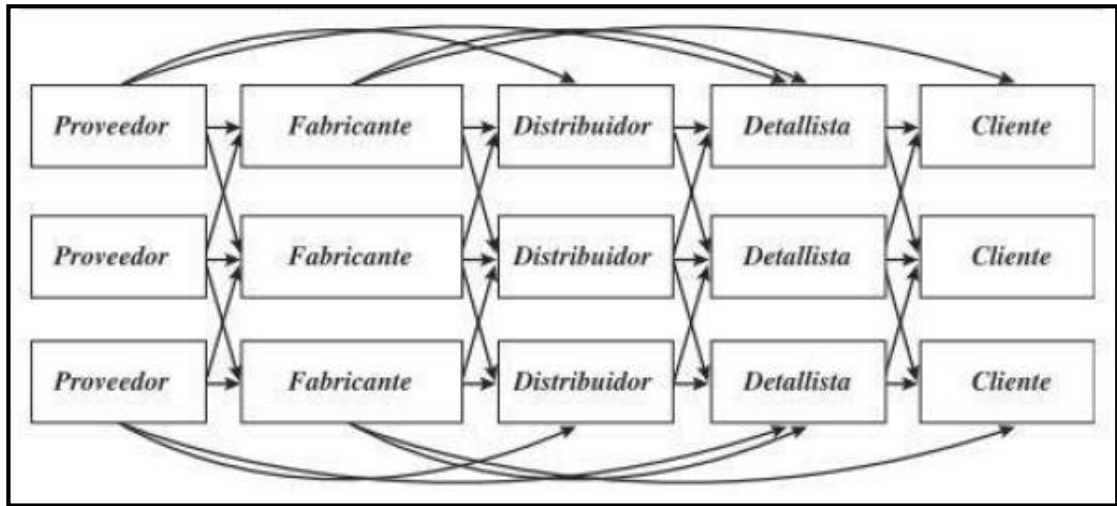
2.2 SISTEMAS LOGÍSTICOS

Un sistema logístico es un conjunto interconectado de recursos, procedimientos y métodos que permiten el sostén logístico. Su misión se sustenta en hacer interactuar de forma ordenada los recursos logísticos para conseguir de modo efectivo los objetivos pronosticados¹⁰.

El sistema logístico de una empresa, en primera instancia, puede esquematizarse por una red, constituida por nodos y arcos para describir la estructura logística en el interior de la fábrica.

¹⁰ Facultad de Ingeniería, Universidad Nacional Autónoma de México. [en Línea] [citado: Diciembre 4, 2013] http://www.ingenieria.unam.mx/industriales/descargas/documentos/catedra/logistica_trabajo.pdf

Ilustración 3 Esquema de la Cadena de Suministro



Fuente: Administración de la Cadena de Suministro, tercera edición, SUNIL CHOPRA, PETER MEINDL

El sistema logístico a su vez es dividido en tres subsistemas que tiene como objetivo general integrar la cadena de suministro¹¹.

- a. Subsistema aprovisionamiento: Incluye los diversos proveedores y comprende todas las operaciones efectuadas para poner a disposición del subsistema producción de las materias primas, las piezas y los elementos comprados. También agrupa las funciones de compras, recepción, almacenamiento y administración de inventarios, e incluye actividades relacionadas con la búsqueda, elección, registro y seguimiento de los proveedores.
- b. Subsistema producción: Transforma la materia prima, efectúa el ensamble de las piezas y los elementos, almacena los productos terminados y los coloca a disposición del subsistema distribución física. Comprende las actividades de mantenimiento y los servicios de planta (suministros de agua, luz, combustibles, etc.), como así también la seguridad industrial y el cuidado del medio ambiente.
- c. Subsistema distribución física: Procede a satisfacer las demandas de los clientes, directamente o mediante almacenes intermedios. ejecuta las actividades de despacho y distribución del producto final a los distintos

¹¹ibid. [en Línea] [citado: Diciembre 4, 2013]

http://www.ingenieria.unam.mx/industriales/descargas/documentos/catedra/logistica_trabajo.pdf

clientes, constituyendo un nexo entre las funciones de producción y de comercialización.

Las actividades logísticas deben coordinarse entre sí para lograr mayor eficiencia en todo el sistema productivo. Por tal motivo, la logística no debe verse como una función aislada, sino como un proceso global de generación de valor para el cliente. Las empresas deben diseñar sus propios procesos con las actividades logísticas (como lo son: aprovisionamiento, compras, almacenamiento, picking, despacho y distribución) que van de acuerdo al servicio que presta y al tipo de clientes que manejan, ofreciendo una mayor velocidad de respuesta con costos mínimos.

2.3 DISTRIBUCIÓN FÍSICA

“La distribución física implica la planeación, la instrumentación y el control del flujo físico de los materiales y los bienes terminados desde su punto de creación/extracción hasta los lugares de su utilización, con el fin de satisfacer las necesidades de los clientes a cambio de una ganancia. El mayor costo de la distribución física corresponde al transporte, seguido por el control de inventario, el almacenaje y la entrega de pedidos con servicios al cliente”¹²

La distribución física puede ser un medidor entre el éxito y el fracaso en los negocios. En esta fase se pueden realizar los ahorros más importantes debido a que el intercambio se facilita por medio de las actividades que ayuden a almacenar, transportar, manipular y procesar pedidos de productos.

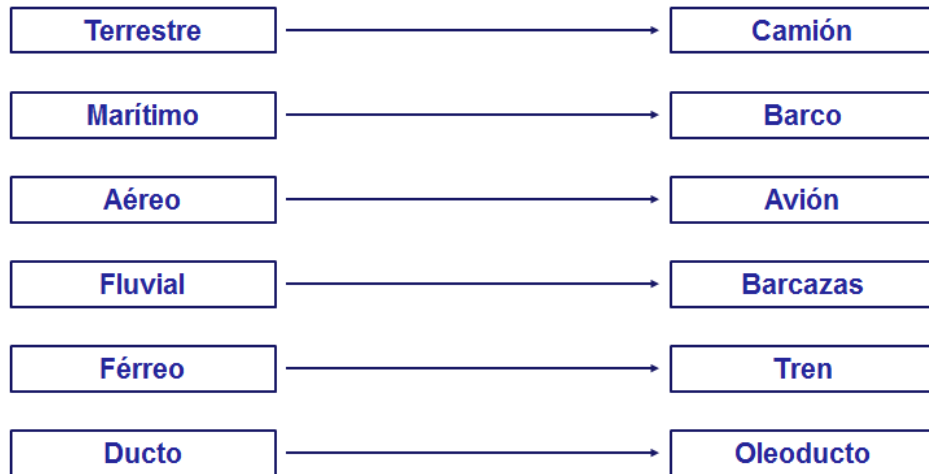
Las decisiones erróneas sobre la distribución física pueden provocar altos costos.

2.3.1 Medios De Transporte

Los medios de transporte son un componente de mucha jerarquía dentro de la distribución física. Para transportar productos dentro y fuera de una ciudad se utiliza las diferentes vías de comunicación. Los transportes más comunes son:

¹² TOLOSA BARÓN, José Luís. Colonia de hormigas fundamentación teórica y aplicación en la optimización de sistemas logísticos de ruteo con intervalos de recepción y tiempo de atención máximo. Tesis de pregrado de Ingeniería Industrial. Bucaramanga. Universidad Industrial de Santander. 2005

Ilustración 4 Medios de Transporte



- Transporte marítimo:

Sus características son:

- ✓ Mayor Capacidad
- ✓ Menor Costo
- ✓ Menor Velocidad
- ✓ Menor Flexibilidad (solo puertos)
- ✓ Está en una fase de crecimiento y mejoramiento debido a los procesos de globalización.

- Transporte Férreo

Sus características son:

- ✓ Flexible en cuanto a tonelaje.
- ✓ Plazos largos de entrega.
- ✓ Especializado en: Lotes de 10 a 20 toneladas, a más de 400 km., si no hay urgencia.

- ✓ Envíos pequeños e irregulares
- Transporte Aéreo

Sus características son:

- ✓ Envíos pequeños
- ✓ Costo elevado
- ✓ Envíos urgentes
- Transporte Terrestre

Sus características son:

- ✓ Muy flexible: plazos, horarios, acceso a los puntos de carga y descarga, personalizado.
- ✓ Factible sin transbordo
- Transporte Fluvial

Sus características son:

- ✓ Bajo costo.
- ✓ Materias pesadas
- ✓ Transporte lento.
- ✓ Factible de formar grandes cargas.

2.3.2 Elección del Medio de Transporte

La elección del medio de transporte está directamente relacionado con las necesidades de los clientes, de acuerdo a¹³:

- a. Determine el tipo de carga que va a transportar: Cada tipo de carga tiene un medio de transporte ideal de acuerdo con sus características físicas y químicas y las restricciones legales existentes

- b. Costo: Se evalúa si los beneficios de uso de un medio de transporte en realidad equivalen a lo que cuesta utilizarlo; en comparación con los demás medios.
- c. Confiabilidad: Luego de tener el medio de transporte de acuerdo con el costo que se puede pagar, el siguiente punto es la determinación de la confiabilidad del medio seleccionado. ¿Qué seguridad tiene mi carga?, ¿Qué manejo se le da?, ¿Operacionalmente es óptimo?
- d. Tiempo de Entrega: El tiempo de entrega va estrechamente ligado a la promesa de cumplimiento a nuestro cliente. Algunos productos no tienen incidencia en la promesa de valor si son entregados hoy o mañana, para muchos otros una diferencia en la hora de entrega puede significar la compra o no del mismo. El tema del tiempo de entrega debe manejarse estratégicamente si se tienen suscritos acuerdos de entregas certificadas.
- e. Asequibilidad: Es la disponibilidad de la ruta o red específica para circular por ella.
- f. Seguridad: La mercancía debe llegar en óptimas condiciones hasta el consumidor, generalmente el transporte se hace responsable de todas las pérdidas y daños en la mercancía.

2.3.3 Canales de Distribución

El canal de distribución lo constituye un grupo de intermediarios relacionados entre sí que hacen llegar los productos y servicios de los fabricantes a los consumidores y usuarios finales.

- Canales de distribución para productos industriales. Los productos industriales tienen una distribución diferente a las de los productos de consumo y emplean cuatro canales que son:
 1. Productores – usuarios industriales: este es el canal más usual para los productos de uso industrial ya que es el más corto y más directo; utiliza representantes de ventas de la propia fábrica.
 2. Productores – distribuidores industriales – consumidores industriales: en este caso los distribuidores industriales realizan las mismas funciones de los mayoristas, en algunas ocasiones desempeñan las funciones de los mayoristas y en otras ocasiones desempeñan las funciones de fuerza de ventas de los fabricantes.
 3. Productores – agentes – distribuidores industriales – usuarios industriales: en este canal la función del agente es facilitar las ventas de los productos y la función del distribuidor es almacenar los productos hasta que son requeridos por el usuario industrial.

4. Productores – agentes – usuarios industriales: en este caso los distribuidores industriales no son necesarios y, por lo tanto, se eliminan. Ejemplo: productos agrícolas.
- Canales de distribución para productos de consumo: Los canales para productos de consumo se dividen a su vez en cinco tipos que se consideran los más usuales:
 1. Productores Consumidores: esta es la vía más corta y rápida que se utiliza en este tipo de productos. La forma que más se utiliza es la venta de puerta a puerta, la venta por correo, el telemercadeo y la venta por teléfono. Los intermediarios quedan fuera de este sistema.
 2. Productores – minoristas – consumidores: este es el canal más visible para el consumidor final y gran número de las compras que efectúa el público en general se realiza a través de este sistema. Ejemplos de este canal de distribución son los concesionarios automotrices, las gasolineras y las tiendas de ropa. En estos casos el productor generalmente hace contacto con los minoristas que venden los productos al público y hacen los pedidos después de lo cual los venden al consumidor final.
 3. Productores – mayoristas – minoristas o detallistas: este tipo de canal se utiliza para distribuir productos tales como medicina, ferretería y alimentos. Se usa con productos de gran demanda ya que los fabricantes no tienen la capacidad de hacer llegar sus productos a todo el mercado consumidor.
 4. Productores – intermediarios – mayoristas – consumidores: este es el canal más largo, se utiliza para distribuir los productos y proporciona una amplia red de contactos; por esa razón, los fabricantes utilizan a los intermediarios o agentes. Esto es muy frecuente en los alimentos perecederos.

2.4 PROBLEMA DE RUTEO DE VEHÍCULOS

En la práctica es necesario aplicar los conocimientos adquiridos a través de la teoría, es importante identificar que la vida está llena de situaciones inciertas regidas por el azar, es por eso que para las empresas de operadores logísticos y/o visitantes aplican para sus recorridos el ruteo vehicular (VRP) basando sus demandas en diferentes tipos.

El origen del VRP data desde el año de 1959 y es introducido por Dantzing y Ramser, quienes describieron una aplicación real acerca de la entrega de gasolina a las estaciones de servicio y propusieron la formulación matemática a este problema. Cinco años después Clarke and Wright propusieron el primer algoritmo

que resultó efectivo para resolverlo (Olivera, 2004). Y es así como se da el comienzo de grandes investigaciones y trabajos en el área de ruteo de vehículos

En inicio el VRP surge como una generalización del problema del agente viajero que se trata de una persona que debe visitar un cierto número de clientes y regresar al lugar de partida, ahora imaginando que se tienen 2 o más agentes viajeros, se llega a la idea de que los clientes pueden ser repartidos entre varios visitadores. El propósito de tener muchos agentes viajeros, suministra a la empresa la posibilidad de ampliar su cartera de clientes pues todos ellos pueden ser visitados por algunos de los agentes.

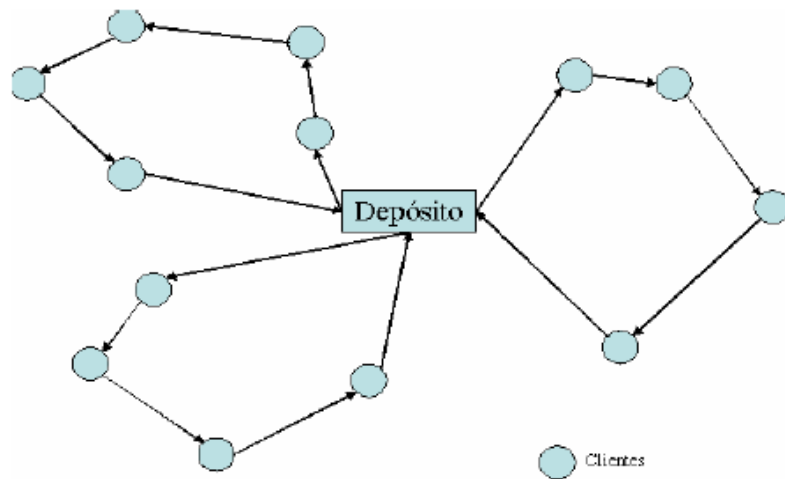
El objetivo del VRP es entregar o recibir recursos a un conjunto de clientes con demandas conocidas o estocásticas, al mínimo costo, hallando las rutas óptimas que empiezan y terminan en un punto.

2.4.1 Definición de un problema de ruteo

El problema del ruteo de vehículos (Vehicle Routing Problem) es ya considerado un paradigma en la literatura especializada. Este problema supone la existencia de un depósito central que cuenta con una flota de vehículos y debe atender a un conjunto de clientes geográficamente dispersos. El objetivo del VRP es entregar bienes a este conjunto de clientes con demandas conocidas, al mínimo costo, encontrando las rutas óptimas que se originan y terminan en el referido depósito. Cada cliente es servido una sola vez, para lo cual se asignan vehículos que llevarán la carga (demanda de los clientes que visitarán) sin exceder su capacidad máxima de transporte¹⁴.

¹⁴ HERMOSILLA, BARÁN, 2001, p.2

Ilustración 5 Representación Gráfica del VRP



Fuente: Laguna (2004)

2.4.2 Variantes del VRP

Los problemas de ruteo de vehículos están definidos entre algunas variantes,¹⁵ las cuales son más comunes las siguientes:

- VRP-TW (Vehicle Routing Problem-Time Windows - Problema con Ventanas Temporales): Se debe encontrar la ruta óptima teniendo en cuenta que cada cliente tiene una ventana horaria [Hora inicio, Hora Fin], y los vehículos solo pueden arribar donde los clientes en el rango de tiempo.
- CVRP (Capacitated Vehicle Routing Problem - Problema con restricciones de Capacidad): Se tiene un conjunto de camiones, los cuales cada uno tienen una capacidad. Entonces se debe encontrar la ruta óptima, recolectando la mercancía de los clientes sin exceder la capacidad del camión.
- CVRPTW (Capacitated Vehicle Routing Problem-Time Windows - Problema con restricciones de Capacidad con Ventanas Temporales): Se debe encontrar la ruta óptima, teniendo en cuenta la capacidad del camión y las ventanas horarias de los clientes.

¹⁵CADENA DE SUMINISTRO [en Línea] [citado: noviembre 2, 2013]

http://www.supplychainw.com/index.php?option=com_content&view=article&id=480:las-variantes-del-problema-de-optimizacion-de-rutas&catid=49:transporte&Itemid=27

- VRPPD (Vehicle Routing Problem with Pick up and Delivery - Problema con Entregas y Devoluciones): Se debe encontrar la ruta optima con la posibilidad de recoger (devoluciones) y entregar mercancía.
- IRP (Inventory Routing Problem – Problema de inventario): Un almacén central con capacidad ilimitada sirve a un conjunto de clientes con cierta demanda fija. El objetivo es resolver un VRP asignando y determinando sus cargas para reponer inventario de los detallistas al mínimo costo de transporte e inventario
- MDVRP (Vehicle Routing Problem with Multiple Depot - Problema con Múltiples Depósitos): En este caso se tienen múltiples depósitos para atender a los clientes, por ende para este problema se necesita asignar a que clientes atiende cada depósito, así como la flota designada a cada depósito.
- SDVRP (Split Delivery Vehicle Routing Problem - Problema de entregas parciales): Para este caso se considera una relajación del problema, en donde se permite que un mismo cliente puede ser servido por diversos vehículos si se reducen los costos totales de operación.
- SVRP (Stochastic VRP - Problema con valores al azar): En esta variante del VRP se considera que algunos de los componentes son estocásticos
- FSMVRPTW (Fleet Size and Mix Vehicle Routing Problem with Time Windows): Es raro encontrar que los vehículos de una flota tengan la misma capacidad, por eso en este caso se considera que los vehículos sean diferentes en cuanto equipo, capacidad de carga, costo por su estructura, y costos de mantenimiento, y además se considera que cada cliente tenga una ventana horaria.

2.4.2 Modelación matemática del VRP.

El problema de ruteo de vehículos es formulado de acuerdo a Marinakis, y Marinaki (2009), de la siguiente manera¹⁶

- Variable de decisión: Es de naturaleza binaria. $x_{ijk}, \forall i, j$

$$1 \leq i, j \leq n + 1, \quad i \neq j, \quad k = 1, \dots, m$$

- ✓ $x_{ijk} = 1$, si en la solución, el camino (i, j) es recorrido por el vehículo k

¹⁶ MARINAKIS, Yannis y MARINAKI, Magdalene. A hybrid genetic – Particle Swarm Optimization Algorithm for the vehicle routing problem. En: Expert Systems with Applications. Marzo, 2009. vol. 37, no, 2., p. 1447.

- ✓ $x_{ijk} = 0$, si en la solución, el camino (i,j) no es recorrido por el vehículo k
- Función objetivo: Se pretende minimizar el costo total de la ruta. La función objetivo es calculada así:

$$\text{Min } z(x) = \sum_{i=1}^{n+1} \sum_{j=1}^{n+1} \sum_{k=1}^m C_{ij} X_{ijk} \quad [1]$$

- Restricciones: Se plantean las siguientes restricciones de conservación del flujo:

$$\checkmark \sum_{k=1}^m \sum_{j=1}^{n+1} X_{ijk} \leq 1 \quad \forall j \in V \setminus \{1\} \quad [2]$$

$$\checkmark \sum_{k=1}^m \sum_{i=1}^{n+1} X_{ijk} \leq 1 \quad \forall i \in V \setminus \{1\} \quad [3]$$

$$\checkmark \sum_{j=1}^{n+1} X_{1jk} = 1 \quad \forall k \quad [4]$$

$$\checkmark \sum_{i=1}^{n+1} X_{i1k} = 1 \quad \forall k \quad [5]$$

$$\checkmark \sum_{i=1}^{n+1} X_{ifk} - \sum_{j=1}^{n+1} X_{fik} = 0 \quad \forall k \quad [6]$$

$$\checkmark X_{ijk} \in \{0,1\} \quad \forall i,j \quad [7]$$

Las restricciones [2] y [3] indican que para cada cliente sólo se puede asignar una ruta vehicular. Las restricciones [4] y [5] limitan el número de rutas por vehículo a uno. [4] hace referencia al número de veces que el vehículo sale del depósito y [5] indica el número de veces que el vehículo llega al depósito. La ecuación [6] determina la continuidad de la ruta, es decir, el número de vehículos que llega a un cliente debe ser igual al número de vehículos que sale. La restricción [7] designa a la variable X_{ijk} los números enteros 0 y 1.

Esta definición del VRP clásica, es un problema idealizado que presenta poca aplicabilidad en escenarios reales, por ello se han incorporado variantes más flexibles, adicionando nuevas restricciones que permitan representar diferentes situaciones prácticas.

2.5 PROBLEMA DE RUTEO DE VEHÍCULOS CON DEMANDAS ESTOCÁSTICAS (VRPSD) Y MC-VRPSD

MC-VRP consiste en el diseño de las rutas de transporte para satisfacer las demandas de un grupo de clientes con un manejo de varios productos, que presenta limitaciones como la incompatibilidad en dichos productos, razón por la cual se deben cargar en los compartimentos de vehículos independientes. Se modela como un programa estocástico con recurso y se resuelve por medio de un algoritmo de memético. Como ejemplo de este tipo de problemas se puede observar en las industrias lácteas ya que estas suelen utilizar vehículos con múltiples compartimentos para recoger la leche de diferentes tipos (por ejemplo, de vacas y cabras) y calidad (por ejemplo, diferentes fechas de succión).

Inicialmente se centró sobre una variante de MC-VRP donde todos los compartimentos de un vehículo pueden ser utilizados para el transporte de cualquier producto. Para resolver el problema, estos autores¹⁷ dividen la solución en dos pasos. En primer lugar, asignar órdenes a los vehículos y los compartimentos, entonces, ellos están al servicio del conjunto de los clientes para solucionarlo en una secuencia del problema del agente viajero (TSP).

El Fallahi¹⁸ propuso un método heurístico basado en el ahorro, una búsqueda tabú (TS), y un algoritmo memético (MA) para una variante MC-VRP en el que cada compartimento se dedica exclusivamente a un producto. Bajo este escenario, no se necesitan las decisiones de asignación ya se le da la capacidad del compartimento para cada producto. Los autores probaron dos políticas de servicio diferentes: en el primero los productos deben ser entregados al cliente por la misma vía, en el segundo cada producto puede ser entregado por una ruta diferente sin embargo, las demandas del cliente son con frecuencia estocásticas, lo que significa que su valor exacto no se conoce en el momento en que se planifican las rutas.

Debido a que con cada producto se modela con una variable aleatoria y con una distribución de probabilidad conocida, los marcos de solución se pueden resolver a partir del problema, estos marcos pueden ser dinámicos o estáticos, Por otra parte, las decisiones de ruteo en los enfoques estáticos, una vez hecho permanecen sin

¹⁷ Mendoza JE, Castanier B, Guéret C, Medaglia AL, Velasco N, A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. Centro para la Optimización y Probabilidad Aplicada (COPA), Industrial Engineering Department, Universidad de los Andes, Bogotá, Colombia, Junio 2009

¹⁸ El Fallahi A, Prins C, Wolfler Calvo R. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research* 2008; 35(5):1725–41.

cambios independientemente de las realizaciones demanda. Desde un punto de vista computacional se prefieren las estrategias estáticas sobre sus contrapartes dinámicas, ya que el problema, conocido por ser computacionalmente intratable, se resuelve una sola vez. Entre los enfoques estáticos, la programación estocástica de dos etapas (o programación estocástica con recurso) ha sido ampliamente utilizada para la VRPSD, un caso particular de la MC-VRPSD en el que cada cliente tiene una demanda estocástica para un solo producto.

El marco de programación estocástica de dos etapas resuelve el problema en dos fases. En la primera fase se ha previsto un conjunto de rutas, mientras que en la segunda fase se ejecutan las rutas planeadas. Sin embargo, debido a la incertidumbre de la demanda, al dar servicio a un cliente determinado, la capacidad restante del vehículo puede no ser suficiente para satisfacer la totalidad de la demanda de los otros clientes, en caso tal, se dice que se produce un fallo de ruta. En caso de un fallo, una acción correctiva predefinida, depende del problema, conocido como recurso y se toma para recuperar la viabilidad de la solución. En general, las fallas de ruta generan un costo adicional cuyo valor sólo se conoce después de la segunda fase. Por lo tanto, el objetivo del problema es diseñar durante la primera fase un conjunto de rutas que minimicen la suma del costo de las rutas planificadas y el costo esperado de las anomalías en la ruta.

En contraste con la VRPSD, la investigación para el MC-VRPSD es escasa. Tatarakis y Minis¹⁹ han abordado una variante de ruta única del problema en el que la secuencia de los clientes se fija de antemano. El problema consiste en la selección de los puntos de reposición de existencias óptimas a lo largo de la ruta. Los autores Tatarakis y Minis²⁰ proponen un conjunto de algoritmos de programación dinámica y resolver los problemas de optimización de hasta 15 clientes.

Para ampliar el conjunto de herramientas disponibles para el MC-VRPSD este documento propone un nuevo algoritmo memético basado en una programación estocástica con la formulación de un recurso para el caso en el que cada compartimento está dedicado a un solo producto. Las parejas de algoritmo han demostrado ser eficaces en los problemas de enrutamiento de vehículos clásicos con un mecanismo adaptado para gestionar la naturaleza estocástica de las demandas en el escenario de múltiples productos y compartimientos. El objetivo es proporcionar un enfoque con una solución alternativa a la heurística de construcción

^{19,20} TATARAKIS A, Minis I. Stochastic single vehicle routing with a predefined customer sequence and multiple depot returns. *European Journal of Operational Research* 2008; 197(2):2.

existente, para hacer frente a casos de gran tamaño actualmente fuera del alcance de métodos exactos.

2.6 ALGORITMOS MEMÉTICOS [MA]

Los Algoritmos Meméticos son algoritmos genéticos (GAs) que recurren a procedimientos de búsqueda local para incrementar la búsqueda²¹. El adjetivo "memético" proviene del término inglés *meme*, acuñado por R. Dawkins²² para designar al análogo del gen en el contexto de la evolución cultural. Resulta conveniente resaltar que la utilización de esta terminología no representa un propósito de adherirse a una metáfora de funcionamiento concreta (la evolución cultural en este caso), sino más bien lo contrario: hacer explícito que se disipa la inspiración puramente biológica, y se opta por modelos más genéricos en los que se manipula, se aprende y se transmite información. En relación con esto último y a la forma en la que más comúnmente un MA puede implementarse, pueden encontrarse diversos trabajos que hacen uso de nombre alternativos para referirse a estos (ej., EAs híbridos), o que aun usando el propio término MA, hacen una interpretación muy restringida del mismo. Puede decirse que un MA es una estrategia de búsqueda en la que una población de agentes optimizadores compiten y cooperan de manera sinérgica. Más aun, estos agentes hacen uso explícito de conocimiento sobre el problema que se pretende resolver, tal como sugiere tanto la teoría como la práctica²³.

El algoritmo memético sigue el siguiente esquema básico:

Generar una población inicial de soluciones

Mejorarlas mediante un método preestablecido UPSO

Repetir

- Seleccionar aleatoriamente un subconjunto de elementos (par) de la población con una probabilidad proporcional a su característica principal.
- Cruce Reproducción: Emparejar o cruzar estas soluciones (padres) para dar lugar a nuevas soluciones (hijos). De cada pareja de padres se han de generar una nueva pareja de hijos.

²¹ MOSCATO P. Memetic algorithms: a short introduction. In: Corne D, Dorigo M, Glover F, editors. *New ideas in optimization*. New York: McGraw-Hill; 1999. p. 219–34.

²² DAWKINS, R.: *The Selfish Gene*. Clarendon Press, Oxford (1976)

²³ CULBERSON, J.: On the futility of blind search: An algorithmic view of "No Free Lunch". *Evolutionary Computation* 6 (1998) 109{127

- Mutación: las soluciones hijas pueden cambiar con una probabilidad pequeña alguno de sus elementos (genes)
- Aplicar el procedimiento de mejora a los hijos
- Sustituir las peores soluciones de la población por las nuevas soluciones hijas

Cabe enfatizar que no existe un procedimiento metódico para afrontar el diseño de un MA en cierto problema ya que podría tener discrepancias con resultados teóricos ya conocidos.

La idea central de los MA está fundamentada en las mejoras individuales de las soluciones de agentes (solución/es al problema asociado al mecanismo de mejora local) que se interrelacionan entre sí en un proceso que contiene fases de cooperación y competiciones del tipo poblacional.²⁴

La fase competitiva es realizada en el proceso de selección y reemplazo, en el primero, varía la distribución de los agentes existentes y el segundo tiene la tarea de limitar el tamaño de la población, es decir, se permite la entrada de nuevos agentes mediante la eliminación de algunos y enfoca la búsqueda.

En la fase de cooperación se crean nuevos agentes, debido que es ejecutada en la aplicación de los operadores de cruce y mutación empleando información extraída de los agentes que han sido recombinados y en algunas ocasiones de información externa (como consecuencia del proceso de mutación).

2.7 ALGORITMO DE OPTIMIZACIÓN DE ENJAMBRE DE PARTÍCULAS (PSO)

Es un método de optimización estocástico basado en la población. Hasta el día de hoy se ha estudiado con éxito en una gran cantidad de problemas de prueba en diversos campos científicos, se cimienta en la observación de la naturaleza, más puntualmente en los movimientos realizados por enjambres de pájaros. Su eficacia se atribuye al intercambio de información entre los puntos de búsqueda que componen la población. Hay dos variantes principales de PSO con respecto al programa de intercambio de información que se utiliza, cada uno con diferentes características de exploración (global) y explotación (local).

²⁴ AGUILAR IMITOLA, Karin Julieth; PEREZ DIAZ, Yuleiny Tatiana. Un algoritmo memético para la minimización del makespan en el problema del job shop scheduling. Tesis de pregrado de Ingeniería Industrial. Bucaramanga. Universidad Industrial de Santander. 2013

PSO es un algoritmo fundado en población, debido a que emplea un enjambre de partículas, que son evaluadas dentro de una zona de exploración por medio de una función objetivo asociada con el problema de optimización al que se quiere dar solución, puede considerarse como un algoritmo sencillo, ya que su foco de procedimientos se halla en el reajuste de la velocidad y posición de cada partícula²⁵.

Los agentes fueron pensados como aves a prueba de colisión, y la idea original era "simular gráficamente la armoniosa pero impredecible coreografía de una bandada". En el documento de Kennedy y Eberhart²⁶ se presentan, las consideraciones generales del método y los pasos a través de los cuales se llegó desde el modelo de aves ya mencionado, al de partículas. Esta es una adaptación resumida que no considera el tamaño de cada partícula sino únicamente su posición.

En su adaptación resumida, el algoritmo consiste en: dada una población inicial, se evalúa en la función objetivo, estableciendo con esto los puntos que pueden ser considerados como la mejor posición (propia) que cada partícula presenta (p_i), y confrontando entre éstas cuál pasa a ser la mejor posición global (p_{gi}). El cambio de posición de cada cliente se determina por una variable (v_i), asociada a la velocidad de cada cliente y que depende de su posición actual, de la mejor posición propia y de la mejor posición global, y que se modela de acuerdo con las siguientes ecuaciones²⁷:

$$v_i^{(k+1)} = X * [v_i^{(k)} + c_1 r_1 (p_i^{(k)} - x_i^{(k)}) + c_2 r_2 (p_{gi}^{(k)} - x_i^{(k)})] \quad (1)$$

$$x_i^{(k+1)} = x_i^{(k)} + v_i^{(k+1)} \quad (2)$$

donde g_i es el índice de la partícula mejor posicionada en la vecindad de x_i ; $i = 1, 2, \dots, N$ número de partículas; k corresponde al contador de iteraciones; X es un parámetro llamado *factor de restricción*, que controla la magnitud de la velocidad; c_1 y c_2 son parámetros positivos de aceleración, llamados parámetro *cognitivo* y *social* respectivamente; y r_1 y r_2 son vectores aleatorios, de valores entre $[0, 1]$. Por el significado de p_{gi} en el algoritmo, se comprende el intercambio de información entre clientes. Para efectuar este algoritmo se siguen cuatro pasos los cuales están puntualizados a continuación²⁸:

²⁵ Efficient Population Utilization Strategy for Particle Swarm Optimizer. Sheng-Ta Hsieh, Tsung-Ying Sun, Chan-Cheng Liu, and Shang-Jeng Tsai. N° 2, APRIL de 2009, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS PART B: CYBERNETICS, Vol. 39, págs. pp. 444 - 456.

²⁶ J. KENNEDY and R. EBERHART, "Particle Swarm Optimization," in Neural Networks, 1995. Proceedings., IEEE International Conference on, 1995, pp. 1942–1948.

²⁷ K. E. PARSOPOULOS and M. N. VRAHATIS, UPSO: A Unified Particle Swarm Optimization Scheme, Lecture series on Computer and Computational Sciences, vol. 1, pp. 868–873, 2004.

²⁸ C. DUARTE and J. QUIROGA, "Algoritmo PSO para identificación de parámetros en un motor DC," Revista Facultad de Ingenierías Universidad de Antioquia, vol. 55, pp. 116–124, 2010.

- a) *Inicialización*: Su objetivo es determinar el intervalo en el cual se encuentran las posiciones iniciales de las partículas. La población inicial se ubica de manera aleatoria en el dominio, la mejor posición p_i inicial, correspondiente al punto más cercano a la solución, es la posición actual. Se determinan también los parámetros de la ecuación.
- b) *Evaluación de la población inicial*: Esta etapa consiste en evaluar todas las partículas, utilizando para ello la función objetivo. De aquí se obtienen los valores de p_i y p_{gi} para el cálculo de las velocidades; donde p_i es la mejor posición de cada partícula y p_{gi} es la mejor posición entre todas.
- c) *Información de posición y velocidad*: Con la ecuación (1) se calcula la velocidad de cada partícula; para luego reubicar cada partícula en el espacio, con la ayuda de la ecuación (2).
- d) *Evaluación de la población actualizada*: Luego de reubicar las partículas, se evalúan de nuevo en la función objetivo para analizar su cercanía a la respuesta (la función objetivo evaluada en las partículas debe tender a $F_{obj} = 0$) de acuerdo con el criterio de parada establecido. Si se cumple, la respuesta final es p_{gi} ; de lo contrario, se realiza de nuevo el paso (c) hasta cumplir con el criterio o abortar el proceso²⁹.

2.8 ALGORITMO DE OPTIMIZACIÓN DE ENJAMBRE DE PARTÍCULAS UNIFICADO (UPSO)

El algoritmo unificado UPSO es un nuevo diseño que tiene en cuenta las variables locales y globales del algoritmo estándar de optimización por enjambre de partículas PSO las cuales son aprovechadas con el fin de que sus características de exploración y explotación sean enlazadas para incrementar así su capacidad de búsqueda del óptimo global. Se ha demostrado que en problemas de tipo multi-objetivos arroja resultados que justifican su superioridad frente a la optimización por enjambre de partículas estándar.

Buscando mejorar la eficiencia del PSO, surgió el UPSO, en el cual se implementa un parámetro global y uno local, así como un parámetro de unificación. Los dos primeros están dados por:

$$G_i^{(k+1)} = X * [v_i^{(k)} + c_1 r_1 (p_i^{(k)} - x_i^{(k)}) + c_2 r_2 (p_g^{(k)} - x_i^{(k)})] \quad (3)$$

$$L_i^{(k+1)} = X * [v_i^{(k)} + c_1 r_1 (p_i^{(k)} - x_i^{(k)}) + c_2 r_2 (p_{gi}^{(k)} - x_i^{(k)})] \quad (4)$$

²⁹ PINZÓN CEBALLOS, Camilo Andrés; ARDILA PEDRAZA, Edwin. Evaluación y comparación de los métodos upso y newton raphson para el análisis de flujo de cargas en un sistema de potencia. Tesis de pregrado de Ingeniería Eléctrica. Bucaramanga. Universidad Industrial de Santander. 2013. Pag 22

Donde, k denota el número de la iteración, g es el índice de la mejor partícula (mejor posición variante global), y gi es el índice de la mejor partícula local (mejor posición variante local). Combinando las ecuaciones (3) y (4) se obtienen las fórmulas de cambio de velocidad y posición respectivamente para UPSO:

$$U_i^{(k+1)} = uG_i^{(k+1)} + (1 - u)L_i^{(k+1)} \quad (5)$$

$$x_i^{(k+1)} = x_i^{(k)} + U_i^{(k+1)} \quad (6)$$

Donde $u \in [0,1]$ es un parámetro llamado factor de unificación, que determina la influencia de los componentes de las variables local y global en la ecuación (5).

Para $u = 1$, la ecuación (5) es igual a la variante global, mientras que para $u = 0$, es igual a la variante local. Para valores $[0,1]$, se obtienen variantes compuestas de PSO que combinan las características antes mencionadas de exploración y explotación de sus variantes³⁰.

La implementación de este algoritmo es similar a la de *PSO*, con la diferencia que ahora se establecerán vecindades de partículas a partir del índice de cada una. Es decir, para establecer una vecindad de, por ejemplo 10 partículas, se ordenan de acuerdo con su índice. Entonces, las partículas de índices $i = 1,2,3, \dots, 10$ forman la primera vecindad, las de índice $= 11,12,13, \dots, 20$ forman la segunda vecindad y así sucesivamente. Luego, de cada vecindad se tomará la partícula con mejor posición (p_{gi}) y de estas posiciones la que esté más cercana a $F_{obj} = 0$ será la mejor posición global (p_g), para luego aplicar las ecuaciones (5) y (6). El factor de unificación (u) es propio de cada caso bajo estudio y se debe buscar el valor que presente una mejor eficiencia de manera empírica; pues aún no hay herramientas matemáticas que ayuden en su búsqueda³¹.

De esta forma se alcanza variables compuestas que fructifican la exploración y explotación de las variables local y global.

³⁰ ibid, pag 24

³¹ Ibid, pag 25

3. DISEÑO DE LA SOLUCIÓN

Existen diferentes técnicas heurísticas para optimizar las rutas de los vehículos basadas en el MC-VRPSD, debido a la escasa información en el tema, diferentes autores proponen distintos puntos de vista con respecto a este tipo de problemas. Tatarakis and Minis³² abordó una variante de ruta única del problema en el que la secuencia de los clientes (o rutas) está fijado de antemano, el problema consiste en seleccionar los puntos de reposición de existencias óptimas a lo largo de la ruta. Mendoza et al³³ propuso un conjunto de construcción heurística que comprenden versiones estocásticas del vecino más cercano y la mejor inserción³⁴. Clarke y Wright³⁵, se adentraron al caso de múltiples compartimientos.

El modelo propuesto plantea no solamente la aplicación de conceptos de los algoritmos UPSO y PSO, y en particular el algoritmo Memético, sino que además integra los conceptos de búsqueda local intensificada en un conjunto de rutas.

El código se desarrolló de la siguiente forma:

a. Datos de entrada para el UPSO

Tabla 1 datos de entrada UPSO

NOMBRE DE VARIABLE	CODIGO
Número máximo de depósitos.	numpar=A
Número de Iteraciones.	numit=B
Factor de Constricción	X=0.5
Parámetro Cognitivo	c1=1.1
Parámetro Social	c2=1.8
Factor de Unificación	u=0.62
Carga variable	Q1 Q2 cargax1=10 cargay1=10
Distancia máxima (metros)	distmax=10

³² TATARAKIS A, Minis I. Stochastic single vehicle routing with a predefined customer sequence and multiple depot returns. *European Journal of Operational Research* 2008;197(2):2.

³³ MENDOZA JE, CASTANIER B, GUÉRET C, MEDAGLIA AL, Velasco N. Approximating the expected cost of recourse on a multi-compartment vehicle routing problem with stochastic demands. Technical Report 08/03/AUTO, Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN). École des Mines de Nantes, France; 2008.

³⁴ BENTLEY JB. Fast algorithms for geometric traveling salesman problem. *INFORMS Journal on Computing* 1992;4:387–411.

³⁵ CLARKE G, Wright JV. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 1964;12(4):568–81.

NOMBRE DE VARIABLE	CODIGO
Distancia mínima (metros)	distmin=5
Producción máxima	Pmax=30
Costo per cápita	cpercapita=3
Velocidad del vehículo (km/h)	velocid=50
Velocidades iniciales (vector)	vin G=vin; L=vin;
Límite superior del rango en donde se generará aleatoriamente la población inicial	ctmaxpromediox1=50
Límite superior del rango en donde se generará aleatoriamente la población inicial	ctmaxpromedioy1=20
Límite inferior del rango en donde se generará aleatoriamente la población inicial	ctminpromediox2=1000
Límite inferior del rango en donde se generará aleatoriamente la población inicial	ctminpromedioy2=1000

b. Vector de Coordenadas que contendrá el número de clientes

En la interfaz del algoritmo se ingresa el valor del número de partículas (*numpar*) las cuales van a ser los clientes (*numpar -1*) y el deposito (*ndeposito*). El valor *numpar* es variable.

```
ndepositos=zeros(numpar,1);
for i=1:numpar
    ndeposito(i,1)=i-1;% Genera el número de clientes junto con el
    deposito dependiendo del numpar
end
length(ndeposito);
```

c. Vector de coordenadas para x.

Se generan las coordenadas en x sujetas al número de clientes (*numpar*) se tiene en cuenta las distancias generadas como valores fijos para efectos del algoritmo (*distmax=10* , *distmin=5*). Los valores de las coordenadas en x son aleatorios.

```
ncoorx=zeros(numpar,1);

for i=1:numpar;
    ncoorx(i,1)=(distmax-distmin)*rand;% Genera las coordenadas en x
    de los clientes.
end
```

```

p=round(ncoorx);%redondea el valor de las coordenadas en x.
% distrestriccion=find(p<distmin);
% for i=1:length(p);
%   if p(i,1)<distmin;
%       p(i,1)=distmin
%   end
%end

length(p);

```

d. Vector de coordenadas para y.

Se generan las coordenadas en y sujetas al número de clientes (*numpar*) se tiene en cuenta las distancias generadas como valores fijos para efectos del algoritmo (*distmax=10* , *distmin=5*). Los valores de las coordenadas en y son aleatorios.

```

ncoory=zeros(numpar,1);

for i=1:numpar;
    ncoory(i,1)=(distmax-distmin)*rand;%Genera las coordenadas en y de
    los clientes.
end
q=round(ncoory);%redondea el valor de las coordenadas en y.
% distrestriccion=find(p<distmin);
%for i=1:length(q);
%   if q(i,1)<distmin;
%       q(i,1)=distmin;
%   end
%end

length(q);

```

e. Calculo de la demanda

La fórmula de la demanda está dada de forma aleatoria basada en el producto del número de clientes (*numpar*) y el valor del producto (*cargax1=10*), los costos máximos del promedio de los productos dado (*ctmaxpromedio_x1=50*, *ctmaxpromedio_y1=20*) y el consumo per cápita estimado para cada clientes (*cpercapita=3*), este es un valor aleatorio para cada cliente. Para mas información de como hallar la demnada potencial ir a las siguiente pagina: [<http://www.crecenegocios.com/como-hallar-la-demanda-potencial/>]

```

d=zeros(numpar,1);
for i=2:numpar

```

```

d(i,1)=((rand*cargax1*length(numpar))/100)+((rand*cargax1)*length(numpar)/100)*(ctmaxpromedioxl*ctmaxpromedioyl)*cpercapita;%Calcula la demanda estimada para cada cliente.
end
D=round(d);%redondea el valor de cada demanda asignada a los clientes.

```

f. Tabla de clientes, coordenadas y demandas.

Se da inicio a la matriz que almacenará los datos del ejercicio de forma resumida. En la columna uno se almacenaran los datos de los clientes incluido el deposito, en la columna dos se mostraran las coordenadas en x, en la columna tres se revelaran las coordenadas de y, y finalmente en la columna cuatro se arrojaran los valores de las demandas. En este ítem se tienen en cuenta la desviación estándar a partir de las demandas de los clientes y a partir de ese valor se obtendrá una demanda estimada que será de apoyo para el cálculo de la capacidad máxima del vehículo y que su carga no supere el volumen permitido a cada compartimiento.

```

tabla=zeros(numpar,4); %Inicia la matriz que almacenará las coordenadas de los clientes junto con la demanda.
for i=1:numpar% For desde 1 hasta la longitud total de numero de clientes.
    tabla(i,1)=ndeposito(i,1); %enumera el numero de clientes.
    tabla(i,2)=p(i,1);%Llena el vector de las coordenadas X.
    tabla(i,3)=q(i,1) ;%Llena el vector de las coordenadas Y.
    tabla(i,4)=D(i,1) ;%Llena el vector de las demandas Y.
end
dt=round(sum(D)/length(D));%valor promedio de la demanda de los vehiculos.
%pdt=dt*Pmax/100;% demanda estimada para cada viaje maxima de cada vehiculo.
DesviacionEs=zeros(length(D),1);%Vector que contendra el valor de las desviaciones a partir de las demandas.
for i=1:length(D)% For desde 1 hasta la longitud total del numero de demandas.
    DesviacionEs(i,1)=(D(i,1)-dt)^2;%obtencion de la desviacion estandar.
end
Des=sqrt(sum(DesviacionEs)/length(D));%valor total de la desviacion estadar.
dt0=Des+dt;%valor de la demanda estimada del vehiculo.
Q=(w*dt0+x1*dt0);%capacidad maxima del vehiculo

```

g. UPSO

Se generan los vectores de la población inicial y de las velocidades iniciales, dependiendo del rango definido por x_{max} y x_{min} para el UPSO y posteriormente se

evalúa la función objetivo inicial para el UPSO con el fin de generar una primera solución para ser evaluada en el ciclo de optimización.

```

%-----
%POBLACIÓN INICIAL UPSO
%-----

xin=zeros(numpar,length(ndeposito)); %Inicia el vector que contendrá la
población inicial.

for i=1:numpar
    for j=1:length(ndeposito)
        xin(i,j)=rand*(distmax-distmin); %Llena aleatoriamente el
vector de posiciones iniciales, dependiendo del rango definido por xmax
y xmin.
    end
end

%-----
%VELOCIDAD INICIAL UPSO
%-----

vin=zeros(numpar,length(ndeposito)); %Inicia el vector que contendrá
las velocidades iniciales de las partículas.

for i=1:numpar
    for j=1:length(ndeposito)
        vin(i,j)=rand*(distmax-distmin); %Llena aleatoriamente el
vector de velocidades iniciales, dependiendo del rango definido por
xmax y xmin.
    end
end

G=vin;
L=vin;

%-----
%EVALUACIÓN INICIAL DE FUNCIÓN OBJETIVO UPSO
%-----

fol=zeros(numpar,1); %Inicia el vector que contendrá los valores de la
evaluación de la función objetivo 1.
TABLAOPTIMIZACION=zeros(numit,2);
tablaxi=zeros(numpar,2); %Inicia la matriz que almacenará las
coordenadas de los clientes junto con la demanda.
for k=1:numpar
    for i=1:numpar% For desde 1 hasta la longitud total de numero de
clientes.
        tablaxi(i,1)=ndeposito(i,1); %numero el numero de clientes.
        tablaxi(i,2)=xin(k,i);%Llena el vector de las coordenadas X.
    end
end

```

h. Calculo de distancias a partir de las coordenadas x,y

La tabla *dist* es generada para poder comparar las distancias que hay entre todos los puntos tanto en el eje x como en el eje y, es una matriz *nxn* idéntica.

```
dist=zeros(numpar+1,numpar+1);%matriz que contendra el valor de las
distancias de cada vehiculo con respecto al deposito.

    for j=2:length(dist)% For desde 1 hasta la longitud total la
matriz de distancias.
        dist(j,1)= ndeposito(j-1,1);%enumera las filas de la matriz
de distancias.
        dist(1,j)=ndeposito(j-1,1);%enumera las columnas de la
matriz de distancias.
    end

for i=2:length(dist)% For desde 1 hasta la longitud total la matriz de
distancias.
    for j=2:length(dist) % For desde 2 hasta la longitud total la
matriz de distancias.
        % dist(i,j)=xin(i-1,j-1)
        dist(i,j)=abs((tablaxi(j-1,2))-(tablaxi(i-1,2)));%comienza a llenar
la matriz con cada valor optenido con las coordenadas x y y
respectivamente de cada cliente.
    end
end
```

i. Calculo de matriz de tiempo

Esta matriz contiene el valor en términos del tiempo que demora en ir de un punto a otros a partir de las distancias arrojadas en la tabla *dist*.

```
velocid=50;%km/h valor asumido de cada vehiculo.

tiempo=zeros(numpar+1,numpar+1);%matriz que contendra el valor de
tiempo a partir de las distancia.

    for j=2:length(tiempo)%For desde 1 hasta la longitud total la matriz
de tiempo.
        tiempo(j,1)= ndeposito(j-1,1);%enumera las filas de la
matriz de distancias.
        tiempo(1,j)=ndeposito(j-1,1);%enumera las columnas de la
matriz de distancias.
    end

for i=2:length(tiempo)%For desde 2 hasta la longitud total la matriz de
tiempo.
    for j=2:length(tiempo)%For desde 2 hasta la longitud total la
matriz de tiempo.
```

```

    tiempo(i,j)=sqrt(((tabla(j-1,3))-(tabla(i-1,3)))^2+((tabla(j-1,4))-(
(tabla(i-1,4)))^2)/velocid;%comienza a llenar la matriz con cada valor
optenido a partir de las distancias de cada cliente.
    end
end

```

j. Asignación de las rutas

Se puede observar el número total de posibles combinaciones dependiendo del número de clientes teniendo en cuenta que el vehículo visitara cuatro clientes por cada ruta es decir va a tener cinco nodos (deposito-cliente1-cliente2-cliente3-cliente4-deposito).

En esta parte de la programación se generan 3 tipos de tablas las cuales guardan la respuesta de las posiciones de los clientes (rutas asignadas = *tabla2*), las demandas de cada cliente respectivamente según la ruta asignada para cada uno (*tabla3*) y las distancias asignadas de cada ruta individualmente (*tabla4*). Con el fin de obtener el valor mínimo de la función objetivo y el índice en donde se localiza de acuerdo a las mejores posiciones encontradas por cada partícula

Las restricciones del problema se muestran como condiciones, las cuales se enfocan en que la demanda del cliente no excede la capacidad maxima del vehículo y en caso de que no cumpla con las condicion se le asigna cero y se descarta esa ruta.

Ver anexo B (código completo)

k. Mejor posición de cada partícula UPSO

Se inicia con el vector que contendrá las mejores posiciones encontradas por cada partícula, para la primera iteración se comienza con las mismas posiciones iniciales, en la segunda iteración se escoge la mejor entre esta y la primera y asi sucesivamente.

```

xbest=zeros(numpar,length(ndeposito)); %Inicia el vector que contendrá
las mejores posiciones encontradas por cada partícula.

for i=1:numpar
    for j=1:length(ndeposito)
        xbest(i,j)=xin(i,j); %Llena el vector de mejores posiciones de
cada partícula con las mismas posiciones iniciales (por lo que es la
primera iteración).
    end
end

```

```
FOxbest=FO;
FOl=sum(FO);
```

I. Mejor posición del enjambre UPSO

De acuerdo a la evaluación generada para cada partícula en el ítem anterior, se evalúa el enjambre o población con el fin de conocer la posición donde se encuentra la mejor solución dada en la función objetivo.

```
xbestglob=zeros(1,length(ndeposito));

[bestglob,ibestglob]=min(FO); %Obtiene el valor mínimo de la función
objetivo y el índice en donde se localiza.
for j=1:length(ndeposito)
    xbestglob(1,j)=xin(ibestglob,j);
end
```

m. Mejor posición del vecindario UPSO

Según el vector que contiene las mejores soluciones globales se evalúan los vecinos con el fin de escoger la mejor partícula (cliente) con el fin de mejorar la función objetivo.

```
xbestveci=zeros(numpar,length(ndeposito)); %Inicia el vector que
contendrá las mejores posiciones encontradas en el vecindario de cada
partícula (vecindario=partícula i-1, i, i+1)

if FO(1,1)<FO(2,1)
    for j=1:length(ndeposito)
        xbestveci(1,j)=xin(1,j); %Si la función objetivo evaluada en la
partícula 1 es menor que la función objetivo evaluada en la partícula
2, la mejor posición del vecindario de la partícula 1 es xin(1,j).
    end
else
    for j=1:length(ndeposito)
        xbestveci(1,j)=xin(2,j); %Si la función objetivo evaluada en la
partícula 1 es mayor que la función objetivo evaluada en la partícula
2, la mejor posición del vecindario de la partícula 1 es xin(2,j).
    end
end

for i=2:(numpar-1)
    if FO(i,1)<FO(i-1,1)&&FO(i,1)<FO(i+1,1)
        for j=1:length(ndeposito)
            xbestveci(i,j)=xin(i,j); %Si la función objetivo evaluada
en la partícula i es menor que la función objetivo evaluada en la
partícula i-1 y en la partícula i+1, la mejor posición del vecindario
es la de la partícula i.
        end
    end
end
```

```

        end
    else
        if FO(i-1,1)<FO(i+1,1)
            for j=1:length(ndeposito)
                xbestveci(i,j)=xin(i-1,j); %Si la mejor posición del
vecindario NO es la de la partícula i, puede ser entonces la de la
partícula i-1 o la de la partícula i+1. Luego, si la función objetivo
evaluada en la partícula i-1 es menor que la función objetivo evaluada
en la partícula i+1, la mejor posición del vecindario es la de la
partícula i-1.
            end
        else
            for j=1:length(ndeposito)
                xbestveci(i,j)=xin(i+1,j); %Si la mejor partícula del
vecindario no es ni la i, ni la i-1, será la i+1.
            end
        end
    end
end
end

if FO(numpar,1)<FO(numpar-1,1)
    for j=1:length(ndeposito)
        xbestveci(numpar,j)=xin(numpar,j); %Si la función objetivo
evaluada en la partícula numpar es menor que la función objetivo
evaluada en la partícula numpar-1, la mejor posición del vecindario de
la partícula numpar es xin(numpar,j).
    end
else
    for j=1:length(ndeposito)
        xbestveci(numpar,j)=xin(numpar-1,j); %Si la función objetivo
evaluada en la partícula numpar es mayor que la función objetivo
evaluada en la partícula numpar-1, la mejor posición del vecindario de
la partícula numpar es xin(numpar-1,j).
    end
end
end

```

n. Ciclo de optimización

En este ciclo se itera de manera constante arrojando varios escenarios los cuales se van cruzando con los datos iniciales presentados en el UPSO eligiendo las mejores soluciones de las variables (padres-hijos). Todos estos datos son visibles en una tabla con las rutas finales (*tabla optimizada*) después de cumplir con las condiciones anteriores de evaluar continuamente hasta que finalice el ciclo de iteraciones de las rutas con el fin de seleccionar la de menor valor. Se evalúan las dos funciones objetivo con los resultados de las mejores posiciones que se encontraron para cada partícula e ir almacenando la respuesta para realizar la sumatoria de la función objetivo final.

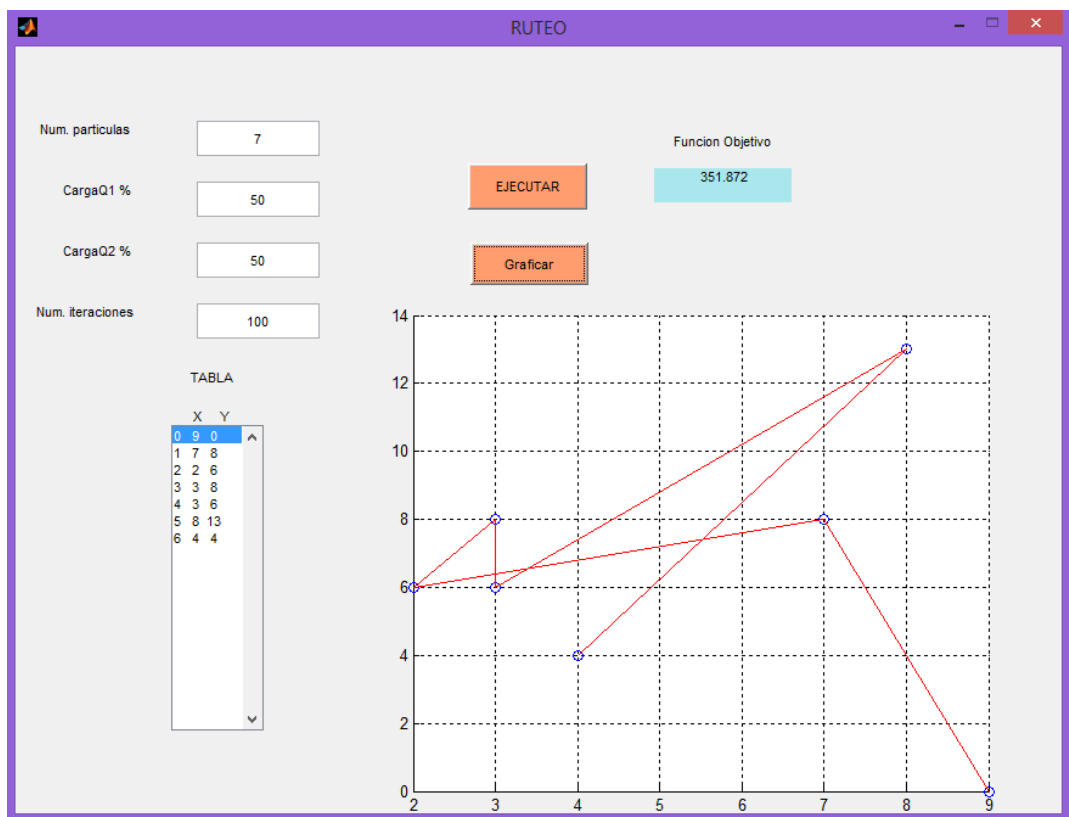
Ver Anexo B (código completo)

3.1 DESARROLLO DEL ALGORITMO COMPUTACIONAL

La aplicación computacional creada en este proyecto fue desarrollada siguiendo el modelo matemático y algorítmico diseñado. Para facilitar su uso, se diseñó una interface de fácil entendimiento pensando en el usuario como un estudioso del algoritmo más que como un usuario de un paquete de ruteo.

El programa fue desarrollado en Matlab.

Ilustración 6 interface del algoritmo computacional creado



4. RESULTADOS COMPUTACIONALES

Ejemplo

Tabla 2 datos ejemplo

NOMBRE DE VARIABLE	CODIGO	VALOR
Número máximo de depósitos.	numpar=A	10
Número de Iteraciones.	numit=B	60
Factor de Constricción	X=0.5	0.5
Parámetro Cognitivo	c1=1.1	1,1
Parámetro Social	c2=1.8	1,8
Factor de Unificación	u=0.62	0,62
Carga variable	Q=q1+q2 cargax1=10 cargay1=10	q1=55 q2=45 Q=100 cargax1=10 cargay1=10
Distancia máxima (metros)	distmax=10	10
Distancia mínima (metros)	distmin=5	5
Producción máxima	Pmax=30	30
Costo per cápita	cpercapita=3	3
Límite superior del rango en donde se generará aleatoriamente la población inicial	ctmaxpromediox1=50	50
Límite superior del rango en donde se generará aleatoriamente la población inicial	ctmaxpromedioy1=20	20
Límite inferior del rango en donde se generará aleatoriamente la población inicial	ctminpromediox2=1000	1000
Límite inferior del rango en donde se generará aleatoriamente la población inicial	ctminpromedioy2=1000	1000

Tabla 3 datos iniciales

NUM CLIENTES	COOR X	COORD Y	DEMANDA
0	4	1	0
1	5	5	11
2	1	5	280
3	5	2	227
4	3	4	118
5	0	1	51
6	1	2	10
7	3	5	14
8	5	4	247
9	5	5	95

Tabla 4 rutas asignadas

Numero de combinaciones 36

RUTAS	Deposito	Cliente 1	Cliente 2	Cliente 3	Cliente 4	Deposito
1	0	1	2	9	8	0
2	0	0	0	0	0	0
3	0	1	4	7	6	0
4	0	0	0	0	0	0
5	0	0	0	0	4	0
6	0	1	7	4	3	0
7	0	0	0	0	2	0
8	0	1	9	2	0	0
9	0	2	3	1	8	0
10	0	2	4	9	7	0
11	0	0	0	0	0	0
12	0	0	0	0	5	0
13	0	2	7	6	4	0
14	0	0	0	0	3	0
15	0	2	9	4	0	0
16	0	3	4	9	8	0
17	0	3	5	8	7	0
18	0	0	0	0	0	0
19	0	3	7	6	5	0
20	0	0	0	0	4	0

RUTAS	Deposito	Ciente 1	Ciente 2	Ciente 3	Ciente 4	Deposito
21	0	3	9	4	0	0
22	0	4	5	9	8	0
23	0	0	0	0	7	0
24	0	4	7	15	6	0
25	0	0	0	0	5	0
26	0	4	9	5	0	0
27	0	0	0	0	8	0
28	0	5	7	8	0	0
29	0	0	0	0	6	0
30	0	5	9	6	0	0
31	0	0	0	0	8	0
32	0	0	0	0	7	0
33	0	0	0	0	0	0
34	0	0	0	0	0	0
35	0	7	9	8	0	0
36	0	0	0	0	0	0

Tabla 5 Demandas de las rutas asignadas

RUTAS	Deposito	Ciente 1	Ciente 2	Ciente 3	Ciente 4	Deposito
1	0	179	21	9	34	0
2	0	179	34	280	246	0
3	0	179	189	34	134	0
4	0	179	134	246	189	0
5	0	179	246	134	34	0
6	0	179	34	189	21	0
7	0	179	280	34	179	0
8	0	179	9	21	0	0
9	0	21	34	179	34	0
10	0	21	189	9	246	0
11	0	21	134	280	134	0
12	0	21	246	34	189	0
13	0	21	34	246	34	0
14	0	21	280	134	21	0
15	0	21	9	189	179	0
16	0	34	189	9	34	0
17	0	34	134	280	246	0
18	0	34	246	34	134	0
19	0	34	34	246	189	0

RUTAS	Deposito	Ciente 1	Ciente 2	Ciente 3	Ciente 4	Deposito
20	0	34	280	134	34	0
21	0	34	9	189	21	0
22	0	189	134	9	34	0
23	0	189	246	280	246	0
24	0	189	34	34	134	0
25	0	189	280	246	189	0
26	0	189	9	134	34	0
27	0	134	246	9	34	0
28	0	134	34	280	246	0
29	0	134	280	34	134	0
30	0	134	9	246	189	0
31	0	246	34	9	34	0
32	0	246	280	280	246	0
33	0	246	9	34	134	0
34	0	34	280	9	34	0
35	0	34	9	280	246	0
36	0	280	9	9	34	0

Tabla 6 matriz de distancias

0	0	1	2	3	4	5	6	7	8	9
0	0	2,0728	0,2383	0,2535	0,1032	0,4818	3,8154	0,8891	0,4815	0,4346
1	2,0728	0	2,3111	2,3263	2,1760	2,5546	1,7426	1,1837	2,5543	2,5074
2	0,2383	2,3111	0	0,0152	0,1350	0,2435	4,0536	1,1274	0,2432	0,1963
3	0,2535	2,3263	0,0152	0	0,1503	0,2283	4,0689	1,1426	0,2280	0,1811
4	0,1032	2,1760	0,1350	0,1503	0	0,3785	3,9186	0,9923	0,3782	0,3314
5	0,4818	2,5546	0,2435	0,2283	0,3785	0	4,2972	1,3709	0,0003	0,0472
6	3,8154	1,7426	4,0536	4,0689	3,9186	4,2972	0	2,9263	4,2969	4,2500
7	0,8891	1,1837	1,1274	1,1426	0,9923	1,3709	2,9263	0	1,3706	1,3237
8	0,4815	2,5543	0,2432	0,2280	0,3782	0,0003	4,2969	1,3706	0	0,0469
9	0,4346	2,5074	0,1963	0,1811	0,3314	0,0472	4,2500	1,3237	0,0469	0

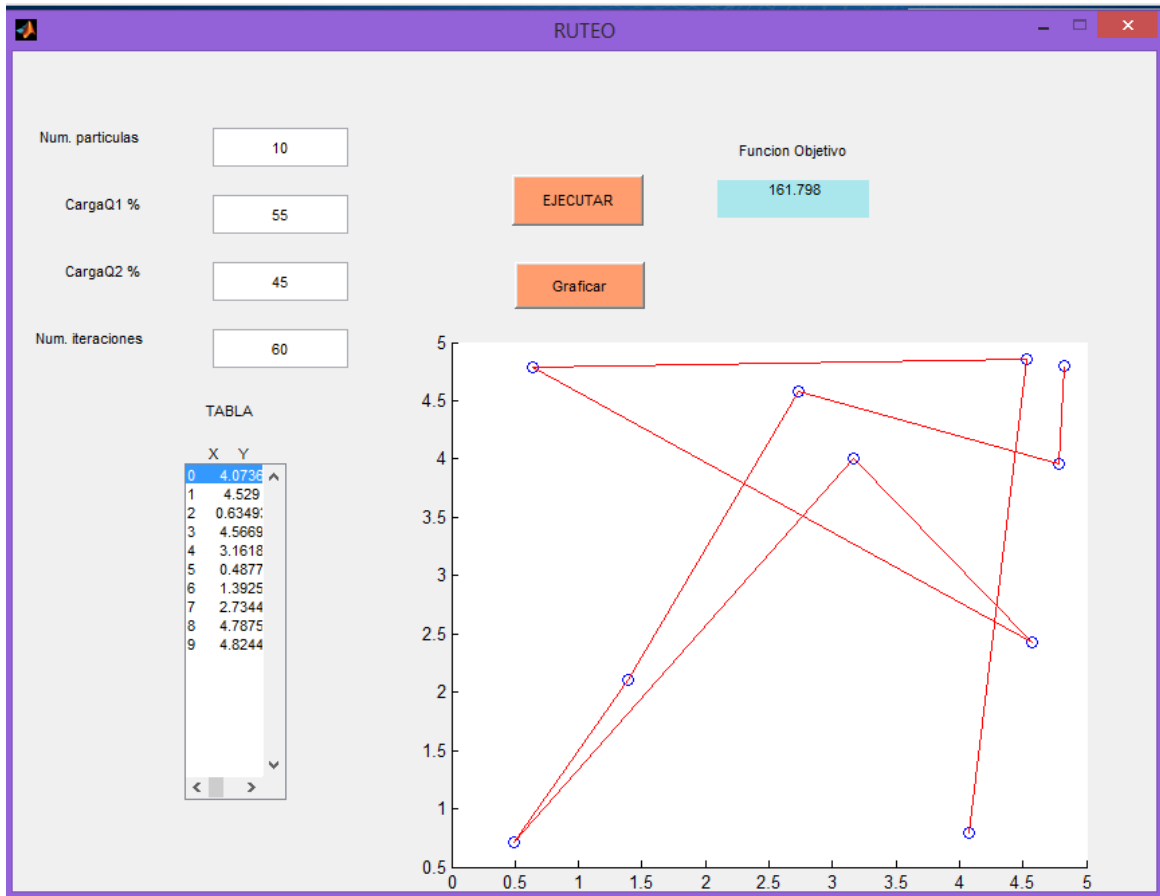
Tabla 7 distancias de las rutas

2,0728	2,3111	0,2383	2,0728	2,3111
0	0	0	0	0
2,0728	2,1760	0,1032	0,2535	2,1760
0	0	0	0	0
0	0	0	0	1,7426
2,0728	1,1837	0,8891	3,8154	1,1837
0	0	0	0	2,5543
2,0728	2,5074	0,4346	0,4815	2,5074
0,2383	0,0152	0,2535	0,0000	0,0152
0,2383	0,1350	0,1032	0,2383	0,1350
0	0	0	0	0
0	0	0	0	4,0536
0,2383	1,1274	0,8891	0,4818	1,1274
0	0	0	0	0,2432
0,2383	0,1963	0,4346	0,8891	0,1963
0,2535	0,1503	0,1032	0,2535	0,1503
0,2535	0,2283	0,4818	0,1032	0,2283
0	0	0	0	4,0689
0,2535	1,1426	0,8891	3,8154	1,1426
0	0	0	0	0,2280
0,2535	0,1811	0,4346	0,4815	0,1811
0,1032	0,3785	0,4818	0,1032	0,3785
0	0	0	0	3,9186
0,1032	0,9923	0,8891	3,8154	0,9923
0	0	0	0	0,3782
0,1032	0,3314	0,4346	0,4815	0,3314
0	0	0	0	4,2972
0,4818	1,3709	0,8891	3,8154	1,3709
0	0	0	0	0,0003
0,4818	0,0472	0,4346	0,4815	0,0472
0	0	0	0	2,9263
0	0	0	0	4,2969
0	0	0	0	4,2500
0	0	0	0	1,3706
0,8891	1,3237	0,4346	0,4815	1,3237
0	0	0	0	0,0469

Q=225

Fobj= 161, 798 mts

Ilustración 7 interface de ejemplo



5. CONCLUSIONES

Se puede observar que la herramienta utilizada para resolver el problema de ruteo de vehículos de múltiples compartimientos con demandas estocásticas MC-VRPSD se resuelve de forma satisfactoria arrojando como resultado la respuesta de solución factible en términos de metros como la ruta más corta.

El algoritmo general usado UPSO, tiene en cuenta a los clientes y el depósito en un espacio aleatorio dentro del área de sondeo y prueba que todos los puntos presentes en el enjambre aportan información favorable para la correlación del método.

Basado en el método de búsqueda local se intensifica la exploración de la ruta óptima una y otra vez con el fin de hacer más eficiente la estrategia genética del algoritmo, demostrando que son más eficientes al momento de trabajar con problemas no lineales y de gran tamaño que los algoritmos evolutivos tradicionales

La representación genética siempre crea rutas y demandas factibles teniendo en cuenta las restricciones de capacidad del vehículo, ruta máxima y mínima por tanto el modo aleatorio permite para crear partículas iniciales, con el propósito de diversificar la población y así no tener resultados sesgados.

BIBLIOGRAFIA

ACOSTA NIÑO, Francia Del Pilar; OSORIO OSORIO, Daniela Angélica. Un método evolutivo de colonia de hormigas para la solución del problema de ruteo de vehículos con demandas estocásticas. Tesis de pregrado de Ingeniería Industrial. Bucaramanga. Universidad Industrial de Santander. 2013.

AGUILAR IMITOLA, Karin Julieth; PEREZ DIAZ, Yuleiny Tatiana. Un algoritmo memético para la minimización del makespan en el problema del job shop scheduling. Tesis de pregrado de Ingeniería Industrial. Bucaramanga. Universidad Industrial de Santander. 2013

C. DUARTE and J. QUIROGA, "Algoritmo PSO para identificación de parámetros en un motor DC," Revista Facultad de Ingenierías Universidad de Antioquia, vol. 55, pp. 116–124, 2010.

CULBERSON, J.: On the futility of blind search: An algorithmic view of "No Free Lunch". *Evolutionary Computation* 6 (1998)

DAWKINS, R.: *The Selfish Gene*. Clarendon Press, Oxford (1976)

NIÑO RUIZ, Elías David, *Optimización Combinatoria: una perspectiva desde la teoría de autómatas*, Editorial Académica Española, Publicado en: Enero 22 de 2012, P 228.

EL FALLAHI A, PRINS C, WOLFLER CALVO R. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research* 2008;35(5):1725–41.

GOMEZ ATUESTA, David Fernando; RANGEL CARVAJAL, Carlos Eduardo. Formular las mateheurísticas Búsqueda Tabú y Reconocido Simulado para la solución del CVRP. Tesis de pregrado de Ingeniería Industrial. Bucaramanga. Universidad Industrial de Santander. 2011

J. KENNEDY and R. EBERHART, "Particle Swarm Optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, 1995, pp. 1942–1948.

K. E. PARSOPOULOS and M. N. VRAHATIS, UPSO: A Unified Particle Swarm Optimization Scheme, Lecture series on Computer and Computational Sciences, vol. 1, pp. 868–873, 2004.

LANGE, Kenneth. Applied Probability: Combinatorial Optimization. En: Springer Science Business Media: Cap. 5 (2010), p. 103.122.

MARINAKIS, Yannis y MARINAKI, Magdalene. A hybrid genetic – Particle Swarm Optimization Algorithm for the vehicle routing problem. En: Expert Systems with Applications. Marzo, 2009. vol. 37, no, 2., p.

MENDOZA JE, CASTANIER B, GUÉRET C, MEDAGLIA AL, Velasco N, A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. Centro para la Optimización y Probabilidad Aplicada (COPA), Industrial Engineering Department, Universidad de los Andes, Bogotá, Colombia, Junio 2009

MOSCATO P. Memetic algorithms: a short introduction. In: Corne D, Dorigo M, Glover F, editors. New ideas in optimization. New York: McGraw-Hill; 1999. p. 219–34.

PINZÓN CEBALLOS, Camilo Andrés; ARDILA PEDRAZA, Edwin. Evaluación y comparación de los métodos upso y newton raphson para el análisis de flujo de cargas en un sistema de potencia. Tesis de pregrado de Ingeniería Eléctrica. Bucaramanga. Universidad Industrial de Santander. 2013. Pag 22-25

TOLOSA BARÓN, José Luis. Colonia de hormigas fundamentación teórica y aplicación en la optimización de sistemas logísticos de ruteo con intervalos de recepción y tiempo de atención máximo. Tesis de pregrado de Ingeniería Industrial. Bucaramanga. Universidad Industrial de Santander. 2005

TATARAKIS A, Minis I. Stochastic single vehicle routing with a predefined customer sequence and multiple depot returns. European Journal of Operational Research 2008; 197(2):2