

**UNA SOLUCIÓN PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS CON
DEMANDA DIFUSA MEDIANTE LA META-HEURÍSTICA DE ENJAMBRE DE
PARTÍCULAS**

ROMÁN ALBERTO ÁLVAREZ CARDONA

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

2015

**UNA SOLUCIÓN PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS CON
DEMANDA DIFUSA MEDIANTE LA META-HEURÍSTICA DE ENJAMBRE DE
PARTÍCULAS.**

ROMÁN ALBERTO ÁLVAREZ CARDONA

**Proyecto de grado para optar al título de
Ingeniero Industrial**

Director

HENRY LAMOS DÍAZ

Ph.D. en Física-Matemática

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

2015

AGRADECIMIENTOS

Agradezco a Dios por haberme permitido alcanzar este logro, por la salud, sabiduría y fortaleza que me brindo durante el transcurso de mis estudios de pregrado y en mi vida personal.

Doy gracias a mis padres Moisés Álvarez y Carmen Cardona, por los valores que me han inculcado además de motivarme a ser mejor cada día. Gracias a mis hermanos Weimar, Heilyn y Jhorman, por apoyarme y porque sé que siempre puedo contar con ustedes. Gracias a mi mascota Homero, por darme afecto, cariño y compañía.

Gracias a mis amigos Oscar Arias y Alexander, por acompañarme, por los gratos momentos que pasamos juntos en la universidad y por los que pasaremos de ahora en adelante.

Gracias al profesor Henry Lamos primeramente por la paciencia que tuvo conmigo, por dirigirme y ayudarme a desarrollar este trabajo y por enseñarme muchas cosas y recordarme muchas más.

Finalmente doy gracias a Tatiana Andrea por ser mi motivación, un modelo a seguir, por apoyarme, aconsejarme, regañarme y por quererme y estar a mi lado en los buenos y malos momentos.

TABLA DE CONTENIDO

	pág.
INTRODUCCIÓN	16
TABLA DE CUMPLIMIENTO DE OBJETIVOS.....	18
1. PLANTEAMIENTO DEL PROBLEMA	19
2. JUSTIFICACIÓN DEL PROYECTO	21
3. RESULTADOS ESPERADOS.....	22
4. OBJETIVOS.....	23
4.1. OBJETIVO GENERAL	23
4.2. OBJETIVOS ESPECÍFICOS.....	23
5. METODOLOGÍA	24
6. REVISIÓN BIBLIOGRÁFICA.....	26
7. FUNDAMENTOS DE LA MATEMÁTICA DIFUSA Y PROGRAMACIÓN LINEAL DIFUSA	32
7.1. INTRODUCCIÓN A LA MATEMÁTICA DIFUSA.....	32
7.2. TEORÍA BÁSICA DE CONJUNTOS DIFUSOS	33
7.2.1. Función de pertenencia y cortes de un conjunto difuso.....	34
7.2.2. Tipos de conjuntos difusos	35
7.2.3. Corte α y corte estricto α	37
7.2.4. Operaciones básicas entre conjuntos difusos	39
7.2.5. Notación para conjuntos difusos discretos o finitos	41
7.2.6. Representación de conjuntos difusos	42
7.2.7. Principio de extensión para los conjuntos difusos	44
7.2.8. Aritmética difusa	44
7.2.9. Variable lingüística.....	46
7.2.10. Operaciones aritméticas en intervalos difusos	48
7.2.11. Operaciones aritméticas entre números difusos.....	50

7.3. PROGRAMACIÓN LINEAL DIFUSA (FLP).....	53
7.3.1. TIPOS DE PROGRAMACIÓN LINEAL DIFUSA.....	53
8. FORMULACIÓN DEL PROBLEMA CVRP CON DEMANDA DIFUSA	59
8.1. CVRP VS FVRP.....	59
8.2. CREDIBILIDAD Y SU APLICACIÓN AL FVRP	63
9. DESCRIPCIÓN DE HEURÍSTICAS Y METAHEURÍSTICAS PARA LA OPTIMIZACIÓN DEL MODELO FVRP	70
9.1. METAHEURÍSTICA DE OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS (PSO)	70
9.1.1. Ecuaciones del PSO.....	71
9.2. HEURÍSTICA DEL VECINO MÁS CERCANO	73
10. CONSTRUCCIÓN DEL FRAMEWORK EN MATLAB DEL MODELO FVRP	75
10.1.DESCRIPCIÓN Y ANÁLISIS DEL ALGORITMO PROPUESTO EN MATLAB	75
10.1.1. Generación de sub-rutas	76
10.1.2. Generación de rutas globales.....	77
10.1.3. Aplicación del pso.....	78
11. MANUAL DE USO DEL FRAMEWORK.....	80
12. DISEÑO DE EXPERIMENTOS DEL MODELO PROGRAMADO	86
13. VERIFICACIÓN DEL MODELO MEDIANTE INSTANCIAS DE BENCHMARKING	92
13.1. ESCENARIO 1 (8 CLIENTES & 1 VEHÍCULO)	94
13.2. ESCENARIO 2 (8 CLIENTES & 2 VEHÍCULOS)	95
13.3. ESCENARIO 3 (14 CLIENTES & 2 VEHÍCULOS)	96
13.4. ESCENARIO 4 (14 CLIENTES & 3 VEHÍCULOS)	97
13.5. ESCENARIO 5 (17 CLIENTES & 3 VEHÍCULOS)	99
13.6. ESCENARIO 6 (17 CLIENTES & 4 VEHÍCULOS)	100
13.7. ESCENARIO 7 (20 CLIENTES & 4 VEHÍCULOS)	102
13.8. ANÁLISIS DE LAS INSTANCIAS.....	104

14. CONCLUSIONES	105
15. RECOMENDACIONES.....	107
BIBLIOGRAFÍA	108
ANEXOS	111

LISTA DE FIGURAS

	pág.
Figura 1. Representación de un conjunto difuso	34
Figura 2. Ejemplo de un conjunto difuso convexo y uno no convexo	36
Figura 3. Ejemplo de dos conjuntos difusos triangulares	37
Figura 4. Complemento estándar entre conjuntos difusos	40
Figura 5. Unión estándar de conjuntos difusos.....	40
Figura 6. Intercepción estándar entre conjuntos difusos	41
Figura 7. Ejemplo de un número difuso y un intervalo difuso	46
Figura 8. Ejemplo de una variable lingüística	47
Figura 9. Número difuso en el caso 1	55
Figura 10. Número difuso triangular para el caso 2.....	57
Figura 11. Red de nodos	68
Figura 12. Resultado de la red	68
Figura 13. Solución por medio del vecino más cercano	74
Figura 14. Sectores del algoritmo programado.....	76
Figura 15. Diagrama de Pareto para los efectos	87
Figura 16. Grafica de normalidad del experimento	88
Figura 17. Grafica de efectos para cada factor.....	89
Figura 18. Grafica de cubos de la interacción de los efectos <i>Cre</i> y <i>w</i>	90
Figura 19. Demanda difusa triangular	93

LISTA DE TABLAS

	pág.
Tabla 1. Ejemplos de operaciones aritméticas.....	49
Tabla 2. Propiedades entre intervalos cerrados.....	49
Tabla 3. Matriz de distancias	73
Tabla 4. Factores del diseño experimental	86
Tabla 5. Datos de entrada para el diseño experimental.....	87
Tabla 6. Contraste de resultados	91
Tabla 7. Posiciones de los nodos escenario 1	94
Tabla 8. Demandas del Escenario 1	94
Tabla 9. Resultado Escenario 1	95
Tabla 10. Posiciones de los nodos escenario 2	95
Tabla 11. Demanda escenario 2	95
Tabla 12. Resultado Escenario 2	96
Tabla 13. Posiciones de los nodos escenario 3	96
Tabla 14. Demanda escenario 3	97
Tabla 15. Resultado Escenario 3	97
Tabla 16. Posiciones de los nodos escenario 4	97
Tabla 17. Demanda escenario 4	98
Tabla 18. Resultado Escenario 4	98
Tabla 19. Distancias entre nodos escenario 5	99
Tabla 20. Demanda escenario 5	99
Tabla 21. Resultados escenario 5.....	100
Tabla 22. Distancias entre nodos escenario 6	100
Tabla 23. Demanda escenario 6	101
Tabla 24. Resultado escenario 6	102
Tabla 25. Distancias entre nodos escenario 7	102

Tabla 26. Demandas Escenario 7	103
Tabla 27. Resultados Escenario 7	103

LISTA DE IMÁGENES

	pág.
Imagen 1. Sectores del modelo programado	79
Imagen 2. Sectores del modelo programado	79
Imagen 3. Parámetros generales del modelo	81
Imagen 4. Matriz de distancias entre nodos	82
Imagen 5. Matriz de demandas Difusas triangulares	83
Imagen 6. Rutas después del PSO.....	83
Imagen 7. Resultados después del PSO	84
Imagen 8. Rutas antes del PSO	85
Imagen 9. Resultados antes del PSO	85
Imagen 10. Parámetros generales de los escenarios.....	93

LISTA DE ANEXOS

	pág.
Anexo A. Programación del framework	111
Anexo B. Artículo.....	134

RESUMEN

TÍTULO: UNA SOLUCIÓN PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS CON DEMANDA DIFUSA MEDIANTE LA METAHEURÍSTICA DE ENJAMBRE DE PARTÍCULAS**

AUTOR: ROMÁN ALBERTO ÁLVAREZ CARDONA**

PALABRAS CLAVES: Problema de ruteo de vehículos con parámetros difusos, Problema de ruteo de vehículos con capacidad, PSO, Enjambre de partículas, Matemática difusa.

CONTENIDO: En el presente documento, se expone un enfoque diferente a la resolución del problema de ruteo de vehículos con capacidad mediante la aplicación de la matemática difusa. Debido al problema que tienen las empresas a la hora de conseguir información válida en lo que se refiere a ciertos parámetros necesarios para la creación de las rutas a usar por sus respectiva flota de vehículos, es necesario explorar nuevas alternativas que logren brindar una solución viable y de calidad en búsqueda de minimizar los costos logísticos generados por el transporte de la mercancía. El concepto de credibilidad en la lógica difusa, puede usarse en conjunto con la programación lineal, como una alternativa válida para el manejo de aquellas situaciones en las cuales alguno de los parámetros de un problema no se conoce con certeza u obtenerlo no es rentable, y aun así, es necesario el embarque de una flota de vehículos que logre cumplir con la demanda de los clientes a un mínimo costo. De tal modo, el objetivo del presente documento es la creación de una herramienta de programación en Matlab®, enfocada a la solución del problema FVRP (Fuzzy vehicle routing problem – Problema de Ruteo de Vehículos Difuso), en el cual la demanda será el factor que se tratará como un número difuso y por consiguiente bajo el concepto de credibilidad. Sin embargo para la creación de dicha herramienta también se darán conceptos básicos de la teoría difusa y una breve explicación de las heurísticas y Metaheurísticas aplicadas al modelo.

* Proyecto de Grado

** Facultad de Ingenierías Físico-Mecánicas, Escuela de Estudios Industriales y Empresariales.
Director PhD Henry Lamos Díaz.

ABSTRACT

TITLE: A SOLUTION TO THE VEHICLE ROUTING PROBLEM WITH FUZZY DEMANDS BY METAHEURISTIC OF PARTICLE SWARM OPTIMIZATION*

AUTHOR: ROMÁN ALBERTO ÁLVAREZ CARDONA**

KEY WORDS: Vehicle routing problem with fuzzy parameters, vehicle routing problem with capacity, PSO, particle swarm, fuzzy mathematics.

CONTENT: In this paper, a different problem solving vehicle routing capacity by applying fuzzy math approach is exposed. Due to the problem that companies have the time to get valid information in regard to certain parameters required for the creation of routes to use for their respective fleet of vehicles it is necessary to explore new ways to achieve a viable and provide quality seeking to minimize logistics costs generated by the transport of the goods. The concept of credibility on fuzzy logic can be used in conjunction with linear programming as a valid alternative for handling situations in which some of the parameters of a problem is not known with certainty or obtain it is not profitable, and still it needed the shipment of a fleet of vehicles which would meet customer demand at minimum cost. Thus, the objective of this document is to create a programming tool in Matlab, focused on solving the problem FVRP (Fuzzy vehicle routing problem - Problem of Fuzzy Vehicle Routing), in which the demand will be the factor which it is treated as a fuzzy number and therefore the concept of credibility. But to create such a tool basics of fuzzy theory and a brief explanation of the heuristics and metaheuristics applied to the model were also given.

* Thesis Project

** Faculty of Physico-Mechanical Engineerings. School of Industrial and Business Studies. Director PhD Henry Lamos Díaz

INTRODUCCIÓN

El incremento en las utilidades y el cumplir con las necesidades del mercado son objetivos claves para el éxito de las empresas. Una forma de alcanzar dichos objetivos es mediante la reducción de costos y el incremento en la productividad en cada una de las dependencias que conforman la industria. La programación lineal ofrece herramientas de apoyo que ayudan a la toma de decisiones, brindando alternativas de solución y manejo a diversos problemas o situaciones que se ubican en el interior de las empresas y que influyen directamente en su desempeño en el mercado, esto mediante la creación de algoritmos matemáticos enfocados al modelamiento de dichos fenómenos.

Gran parte del costo logístico en las empresas es generado por el transporte del producto al cliente, es así que, con el paso de los años esta situación se ha transformado en un caso de estudio en la ingeniería industrial. Uno de los problemas básicos de transporte es conocido como el “CVRP” (*Problema de Ruteo con Capacidad - Capacited Vehicle Routing Problem*), el cual consiste en encontrar el conjunto de rutas que minimicen en lo posible las distancias recorridas (costos de transporte) por una flota de vehículos que tienen una capacidad finita Q (por ejemplo, toneladas). Con el objetivo de capturar situaciones más reales se han desarrollado variantes del problema básico de ruteo; una de éstas es cuando se considera que los parámetros del problema son difusos FVRP (por sus siglas en inglés *Fuzzy Vehicle Routing Problem – Problema de ruteo de vehículos difuso*).

En el presente trabajo se considera la demanda de los clientes como un número difuso, esto como reflejo de la dificultad en determinar la demanda de manera exacta en algunas ocasiones, bien sea por costo o falta de información. El CVRP es un problema que pertenece a la clase *NP hard* (*Non-deterministic Polynomial-*

time hard), esto significa que la solución para situaciones en la que el número de clientes es relativamente grande, es difícil de obtener por medio de algoritmos exactos, por consiguiente, en su lugar lo que se hace es determinar soluciones factibles buenas mediante algoritmos con un bajo costo computacional, estos algoritmos se denominan “Heurísticos”. El presente trabajo propone hallar una solución factible para el problema FVRP usando la meta-heurística PSO (*Particle swarm optimization* - Optimización por Enjambre de Partículas), que se validará mediante instancias de la literatura, y de este modo construir una herramienta software que sea útil para su implementación en la industria.

TABLA DE CUMPLIMIENTO DE OBJETIVOS

OBJETIVO	CUMPLIMIENTO
Realizar una revisión bibliográfica en base de datos científicas para el problema de ruteo de vehículos con capacidad con parámetros difusos.	Capítulo 6
Construir un documento introductorio sobre matemática difusa y su aplicación en problemas de optimización.	Capítulo 7
Elaborar un framework para el problema CVRP con demanda difusa mediante la herramienta software Matlab®.	Capítulo 10
Describir los procedimientos en un manual para el manejo de la herramienta diseñada.	Capítulo 11
Evaluar la calidad de la solución obtenida por el algoritmo mediante instancias.	Capítulo 13
Elaborar un artículo de carácter publicable con los análisis, resultados y conclusiones del tema estudiado.	Anexo 2

1. PLANTEAMIENTO DEL PROBLEMA

La investigación de operaciones es una rama de las matemáticas cuyo objetivo es el desarrollo de modelos de optimización en búsqueda de soluciones prácticas y eficientes para problemas en la tanto para la industria como para diferentes situaciones de la vida cotidiana. El problema VRP (*Vehicle Routing Problem* - Problema de Ruteo de Vehículos) desde hace ya algún tiempo es objeto de estudio por ingenieros e investigadores, debido a la gran cantidad de variantes que existen de este tipo de problemas y la complejidad de ellos.

El CVRP es uno de los ejemplos más sencillos de los problemas de ruteo, el cual consiste en satisfacer las demandas de un conjunto de clientes distribuidos en una zona geográfica por una flota de vehículos con una capacidad definida (Q), que inician y finalizan su recorrido en un sitio determinado, el cual es llamado deposito. Sin embargo, en este tipo de problema la demanda del cliente entra a tomar importancia, a causa de que no todas las empresas logran tener los datos exactos de la demanda a tiempo (antes de construir las rutas a utilizar). Todo esto puede llegar a afectar a la organización elevando los costos de transporte debido al uso de un sistema de rutas “no óptimas” para el problema.

*Peng y Qian*¹ exponen un modelo para el FVRP cuya demanda es manejada como un valor difuso. A su vez, este modelo es resuelto por los mismos utilizando la metaheurística de Enjambre de Partículas (PSO). Este modelo puede dar un manejo a diferentes problemas que se evidencian en el día a día en las industrias y a nuestro alrededor; como la recolección de basuras, el reabastecimiento de cajeros automáticos o estaciones de gasolina, la distribución de productos tienda a

¹ PENG, Yang y QIAN, Ye-mei. A particle swarm optimization to vehicle routing problem with fuzzy demand. En: Convergence Information Technology. Agosto, 2010. vol. 5, no. 6, p. 112-119

tienda, la entrega de correos, etc. El proyecto a realizar, propone la construcción de una herramienta software basada en el modelo FVRP expuesto, buscando dar un uso apropiado y eficiente al mismo.

La lógica difusa es una extensión de la matemática tradicional y es considerada hoy en día como una parte fundamental del desarrollo de la inteligencia artificial, dado que los valores de cierta variable no se consideran exactos, si no que se usan niveles y grados de pertenencia, los cuales están más acorde con el pensamiento humano. Por otro lado, la lógica difusa es cada día más utilizada en las industrias para poder dar solución a gran variedad de problemas que la matemática tradicional determinística no logra resolver de manera óptima, de esta forma es como este tema cobra importancia en la investigación de operaciones y en este caso específicamente, en los problemas de ruteo de vehículos.

2. JUSTIFICACIÓN DEL PROYECTO

El problema del transporte de carga ocasiona grandes costos a las empresas comparados con el costo logístico total, en conjunto con un mercado competitivo el cual exige entregas de pedidos a tiempo de manera eficiente, hacen de éste problema una necesidad de estudio para investigadores y organizaciones. Es así como la implementación de modelos de programación lineal enfocados hacia la optimización de las rutas y por tanto la minimización de los costos en este campo es de vital importancia, así como la construcción e implementación de herramientas informáticas que faciliten su correcta implementación en la industria.

El proyecto propuesto se fundamenta en tres factores encaminados hacia la búsqueda de una solución factible para el CVRP aplicando conceptos de lógica difusa, con el fin de elaborar una herramienta práctica para su uso en el mundo empresarial. Para ejecución de la propuesta se partirá de una revisión bibliográfica con el fin de documentar científicamente el problema de ruteo de vehículos con parámetros difusos y de esta manera alimentar la base de datos del grupo de investigación OPALO. A partir de esta documentación, se realizará una breve introducción a la lógica difusa y cómo ésta puede ser utilizada para dar mejores soluciones al problema de ruteo. Finalmente, se buscará dar una solución al problema FVRP, mediante la creación de una framework en el software Matlab®, el cual posteriormente se someterá a una validación mediante instancias para verificar la calidad de la solución generada y del mismo.

3. RESULTADOS ESPERADOS

Los resultados esperados de la investigación basados en los objetivos planteados en el proyecto propuesto son los siguientes:

- Presentar un documento en el cual se muestre de una manera ordenada la bibliografía enfocada al uso de la matemática difusa en problemas de optimización, programación lineal, programación entera y de ruteo de vehículos.
- Framework del problema CVRP con demanda difusa.
- Implementación del algoritmo en el lenguaje de Programación Matlab® y su validación mediante instancias.
- Un artículo de carácter publicable en el cual se documente el análisis, los resultados y las conclusiones del tema expuesto en el presente proyecto.

4. OBJETIVOS

4.1. OBJETIVO GENERAL

Resolver el problema de ruteo de vehículos con demanda difusa por medio de la metaheurística de enjambre de partículas.

4.2. OBJETIVOS ESPECÍFICOS

- Realizar una revisión bibliográfica en base de datos científicas para el problema de ruteo de vehículos con capacidad con parámetros difusos.
- Construir un documento introductorio sobre matemática difusa y su aplicación en problemas de optimización.
- Elaborar un framework para el problema CVRP con demanda difusa mediante la herramienta software Matlab®.
- Describir los procedimientos en un manual para el manejo de la herramienta diseñada.
- Evaluar la calidad de la solución obtenida por el algoritmo mediante instancias.
- Elaborar un artículo de carácter publicable con los análisis, resultados y conclusiones del tema estudiado.

5. METODOLOGÍA

Durante el transcurso y para el desarrollo del proyecto propuesto se realizarán las actividades que se muestran a continuación:

- *Revisión Bibliográfica:*

Realizar una búsqueda en las bases de datos con temas multidisciplinares como EISEvier, Scopus, Science Direct, SpringerLink entre otras y seleccionar la bibliografía disponible acerca del uso de la matemática difusa en los problemas de optimización, programación lineal, programación entera y el problema de ruteo de vehículos. Posteriormente, se recopilará la información en un documento de tal manera que sirva para documentar científicamente el CVRP con demanda difusa.

- *Introducción a la Matemática Difusa y Programación Lineal con Parámetros Difusos:*

En el documento se presenta una breve introducción a la matemática difusa desde la perspectiva de conjuntos y posteriormente la formulación matemática de los problemas de programación lineal difusa.

- *Construcción del Modelo CVRP con demanda difusa:*

Se construye un modelo matemático para el problema CVRP con demanda difusa en base al artículo "A particle swarm optimization to vehicle routing problem with fuzzy demand", publicado en la revista "Convergence Information Technology" volumen 5 en agosto del 2010 por los investigadores Yang Peng y Ye-mei Qian.

- *Construcción del Framework del CVRP con demanda difusa*

- *Implementación del algoritmo en Matlab®:*
Se elabora un pseudocódigo en el lenguaje de Matlab® del modelo CVRP con demanda difusa propuesto con anterioridad.
- *Validación:*
Se seleccionan algunos escenarios para el problema CVRP con demanda difusa y posteriormente, se utilizan para observar los resultados obtenidos.
- *Diseño Experimental:*
Se consultan diversos tipos de diseños de experimentos y se selecciona el más apropiado para la determinación de los factores que influyen en la calidad de la respuesta del modelo CVRP con demanda difusa propuesto. Se siguen las pautas de acuerdo al diseño, identificando las variables y el nivel de confianza determinando de esta forma el número de réplicas necesarias.

6. REVISIÓN BIBLIOGRÁFICA

El estudio de la matemática difusa se inició en el año 1965 siendo propuesta por el matemático, ingeniero eléctrico, informático y profesor azerbaiyano Lotfi Asker Zadeh. El ingeniero *Lotfi*, se introduce al mundo la teoría de conjuntos difusos o lógica difusa, esto se da en su afán por elaborar una herramienta que facilitara y diera un mejor manejo a sistemas complejos en la industria generados por una ambigüedad en la estructura de sus datos, para los cuales la matemática tradicional dada su exactitud no podía dar una solución fiable.

En el transcurso de los ya casi 50 años de estudio de esta temática, la matemática difusa ha permitido modelar diferentes fenómenos de la industria, en lo que respecta a su aplicación en procesos de toma de decisiones en las empresas y la robótica, hasta la predicción de desastres naturales. Para el presente trabajo serán relevantes aquellas aplicaciones que se dan de la matemática difusa a problemas de optimización, programación lineal, programación entera y problemas de ruteo de vehículos, de este modo, a continuación se presentará una bibliografía enfocada a estas aplicaciones en los últimos años.

Muchos problemas en la industria no son fáciles de modelar debido, en primer lugar, a la variabilidad presentada por los datos experimentales o por la necesidad de aumentar el espectro de los resultados obtenidos por medio de estos. Problemas de optimización que han sido tratados mediante el uso de la matemática difusa se describen a continuación:

Tan y otros² proponen y desarrollan un modelo para la optimización de la producción de biomasa y el comercio en función de la disponibilidad de recursos y las limitaciones provocadas por la huella ambiental, a partir de un modelo de optimización de entrada-salida multi-regional con parámetros difusos.

Aviso y otros³ presentan un modelo de optimización entrada-salida difusa enfocado a la optimización y mejora de las cadenas de suministro ecológico-industriales bajo las limitaciones de la huella hídrica estudiando los casos de la fabricación de azulejos y la producción de biocombustibles.

De Oña y otros⁴ introducen un nuevo método para utilizar la optimización difusa en el ajuste de los datos de valores observados de manera que cumplan con las ecuaciones de conservación de flujo y cualquier requisito de consistencia. La novedad radica en la posibilidad de obtener la mejor combinación de los valores ajustados, preservando así la integridad de datos tanto como sea posible.

Selim y otros⁵ proponen un manejo para la planificación de la producción-distribución colaborativa en la cadena de suministro mediante programación lineal planteando metas u objetivos difusos, de este modo aprovechar la variabilidad que

² TAN, Raymond R.; AVISO, Kathleen B.; BARILEA, Ivan U.; CULABA, Alvin B.; CRUZ JR, Jose B. A fuzzy multi-regional input-output optimization model for biomass production and trade under resource and footprint constraints. En: Applied Energy. Febrero, 2012. vol. 90, p. 154-160.

³ AVISO, Kathleen B.; TAN, Raymond R.; CULABA, Alvin B.; CRUZ JR, Jose B. Fuzzy input e output model for optimizing eco-industrial supply chains under water footprint constraints. En: Journal of Cleaner Production. Enero-Febrero, 2011. vol. 29, no. 2-3, p. 187-196.

⁴ DE OÑA, Juan; GÓMEZ, Penélope; MÉRIDA-CASERMEIRO, Enrique. Bilevel fuzzy optimization to pre-process traffic data to satisfy the law of flow conservation. En: Transportation Research Part C. Emerging Technologies. Febrero, 2011. vol. 19, no. 1, p. 29-39.

⁵ SELIM, Hasan; ARAZ, Ceyhun; OZKARAHAN, Irem. Collaborative production-distribution planning in supply chain: A fuzzy goal programming approach. En: Transportation Research Part E: Logistics and Transportation Review. Mayo, 2008. vol. 44, no. 3, p. 396-419.

proporciona la teoría difusa para en primer lugar plantear un modelo más realista y en segundo lugar buscar obtener mejores soluciones para este tipo de problemas.

Los siguientes artículos exponen modelos de programación lineal y programación entera con parámetros difusos:

Eshghi y Nematian⁶ exponen enfoques difusos a diversos problemas de programación entera, desde un modelo principal en el cual exponen las limitantes del mismo hasta al final por medio del empleo de la matemática difusa en conjunto con técnica de descomposición, el método de ponderación y el algoritmo *Branch and Bound* pueden dar un mejor manejo y dar una solución más favorable al modelo propuesto.

Shivaie y otros⁷ proponen un modelo de programación entera no lineal aplicando conceptos de conjuntos difusos en el cual los objetivos son la minimización de los costos de inversión y el aumento de los beneficios para el cliente. El modelo propuesto se da con el fin de construir un mecanismo que sirva para la toma de decisiones fundamentadas en los conceptos de rentabilidad y mercado.

Nasrabadi y otros⁸ presentan un modelo de aproximación para los problemas separables de programación lineal continua cuya función objetivo está construida mediante la aplicación de conceptos de matemática difusa, con el fin de tratar

⁶ ESHGHI, Kourosch y NEMATIAN, Javad. Special classes of fuzzy integer programming models with all-different constraints. En: Scientia Iranica. Enero-Junio 2009. vol. 16, no. 1, p. 1-10.

⁷ SHIVAIE, M., SEPASIAN, M.S. y SHEIKH-EL-ESLAMI, M.K. Multi-Objective transmission expansion planning using Fuzzy-Genetic Algorithm en Iranian Journal of Science and Technology. En: Transaction B: Engineering. Enero, 2011. vol. 35, no. 2, p. 141-159.

⁸ NASRABADI, M.M., YAGHOobi, M.A. y MASHINCHI, M. Separated continuous linear programs with fuzzy valued objective function. En: Scientia Iranica. Julio-Diciembre, 2010. vol. 17, no. 2, p. 105-118.

estos modelos cuyos parámetros de decisión tienen una variabilidad derivada de las situación de las cuales se obtienen.

Por último se presentan los artículos de mayor relevancia para el proyecto en desarrollo. Los siguientes artículos exponen aplicaciones de matemática difusa en diferentes variantes del problema de ruteo de vehículos:

Mahdieh y otros⁹ proponen una serie de modelos selectivos de optimización SVRP (*Selective Vehicle Routing Problem – Problema de ruteo de vehículos selectivo*), entre los cuales proponen un FSVRP (*Fuzzy Selective Vehicle Routing Problem – problema de ruteo de vehículos selectivo difuso*) forma difusa del SVRP, los cuales se encuentran enfocados a maximizar la fiabilidad, la robustez y la importancia de la solución. Exponen la importancia del FSVRP en la gestión de desastres debido al manejo que es posible dar a la incertidumbre por medio de este modelo.

Erbao y Mingyong¹⁰ abordan a partir del problema OVRP (*Open Vehicle Routing Problem – Problema de ruteo de vehículos abierto*), un nuevo modelo denominado OVRPFD (*Open Vehicle Routing Problem with Fuzzy Demands – Problema de ruteo de vehículos con demanda difusa y abierto*) manejando el parámetro de la demanda como un número difuso. A demás propone una solución mediante la creación de un algoritmo híbrido inteligente entre simulación estocástica y evolución diferencial.

⁹ ALLAHVIRANLOO, Mahdieh; CHOW, Joseph. YJ. y RECKER, Will W. Selective vehicle routing problems under uncertainty without recourse. En: Transportation Research Part E: Logistics and Transportation Review. Febrero, 2014. vol. 62, p. 68-88.

¹⁰ ERBAO, Cao y MINGYONG, Lai. A hybrid differential evolution algorithm to vehicle routing problem with fuzzy demands. En: Journal of Computational and Applied Mathematics. Septiembre, 2009. vol. 231, no. 1, p. 302-310.

Caggiani y otros¹¹ introducen la matemática difusa a los problemas Destino-Origen (O-D) manejados mediante GLS (*Generalised Least Square estimators – Estimadores de mínimos cuadrados generalizados*) proponen un nuevo modelo denominado Fuzzy-GLS en búsqueda de mejorar las prestaciones de estimación incluyendo conteos de tráfico e información de cómo iniciar la demanda incierta O-D.

Liu y otros¹² aplica conceptos de matemática difusa a los problemas de ruteo de vehículos cuya demanda es incierta en los nodos del modelo, con el fin de obtener un mejor rendimiento en los resultados arrojados mediante la simulación del problema.

Xu y otros¹³ propone una solución del modelo de ruteo de vehículos con ventanas de tiempo suaves en un entorno aleatorio difuso al minimizar la distancia recorrida por la flota de vehículos y a su vez maximizar el nivel medio de satisfacción de los clientes servidos.

¹¹ CAGGIANI, Leonardo; OTTOMANELLI, Michele y SASSANELLI, Domenico. A fixed point approach to origin-destination matrices estimation using uncertain data and fuzzy programming on congested networks. En: Transportation Research Part C: Emerging Technologies. Marzo, 2013. vol. 28, p. 130-141.

¹² LIU, Chang-Shi y LAI, Ming-Yong. The vehicle routing problem with uncertain demand at nodes. En: Transportation Research Part E: Logistics and Transportation. Julio, 2009. vol. 45, no. 4, p. 517-524.

¹³ XU, Jiuping; YAN, Fang y LI, Steven. Vehicle routing optimization with soft time windows in a fuzzy random environment. En: Transportation Research Part E: Logistics and Transportation Review. Noviembre, 2011. vol. 47, no. 6, p. 1075-1091.

Ghaffari-Nasab y otros¹⁴ introduce demandas difusas al problema de localización-ruteo y propone una solución para el modelo para dar un mejor manejo logístico a varios problemas del día a día en las industrias.

Zheng y Liu¹⁵ construyeron un algoritmo híbrido inteligente para un problema difuso de ruteo de vehículos. Además introducen la credibilidad difusa a usar en estos modelos.

¹⁴ GHAFFARI-NASAB, N.; GHAZANFAR AHARI, S. y GHAZANFARI, M. A hybrid simulated annealing based heuristic for solving the location-routing problem with fuzzy demands. En: Scientia Iranica. Junio, 2013. vol. 20, no. 3, p. 919-930.

¹⁵ ZHENG, Yongshuang y LIU, Baoding. Fuzzy vehicle routing model with credibility measure and its hybrid intelligent algorithm. En: Applied Mathematics and Computation. Mayo, 2006. vol. 176, no. 2, p. 673-683.

7. FUNDAMENTOS DE LA MATEMÁTICA DIFUSA Y PROGRAMACIÓN LINEAL DIFUSA

7.1. INTRODUCCIÓN A LA MATEMÁTICA DIFUSA

La matemática difusa o borrosa, es una rama de la matemática clásica que nace a partir de la necesidad de cuantificar estados tales como “muy caliente”, “poco alto”, “muy bonita” etc., los cuales no es posible describir por medio de los conjuntos clásicos o *crisp*, dado que estos últimos cumplen una función de incluir o excluir a los miembros de una población A transformándolos en los que “pertenecen” y los que “no pertenecen” a dicho conjunto.

A través de los últimos años, la matemática difusa se ha podido aplicar en diferentes sectores de la industria, buscando dar soluciones más reales a problemas de alta complejidad, los cuales requieren un tratamiento diferente al que se le da mediante la matemática clásica.

Siendo la matemática difusa una parte de la matemática tradicional clásica, se debe considerar que ésta se encuentra del mismo modo fundamentada en su propia teoría de conjuntos. Por consiguiente, para el desarrollo del presente trabajo se darán conceptos básicos de teoría de conjuntos difusos y aritmética difusa como resultado del estudio y aprendizaje extraídos del libro de Cano¹⁶.

¹⁶ CANO, Yurilev Chalco. Algunas contribuciones a la teoría difusa multivariable. Febrero, 2004. Universidad Estatal de Campiñas. p. 1-10.

7.2. TEORÍA BÁSICA DE CONJUNTOS DIFUSOS

En la teoría clásica de conjuntos, las funciones para estos son de tipo incluyente y excluyente, es decir, separan los individuos del universo en el cual se encuentra incluido el conjunto, en dos grupos formalmente llamados “los que pertenece y los que no pertenecen”. La función de pertenencia para un conjunto *crisp* se define de la siguiente forma:

$$B(x) = \begin{cases} 1 & \text{si pertenece} \\ 0 & \text{si no pertenece} \end{cases}$$

En la teoría de conjuntos difusos, la función del conjunto es denominada como su “Función de membresía o pertenencia” y esta última se diferencia de la teoría clásica de conjuntos, dado que no es una función delimitadora excluyente ni incluyente por medio de los extremos de conjunto, sino que se referirá a un grado o nivel de pertenencia simbolizado comúnmente por la letra α , que se usa para representar un grado de pertenencia de un individuo a un conjunto difuso específico.

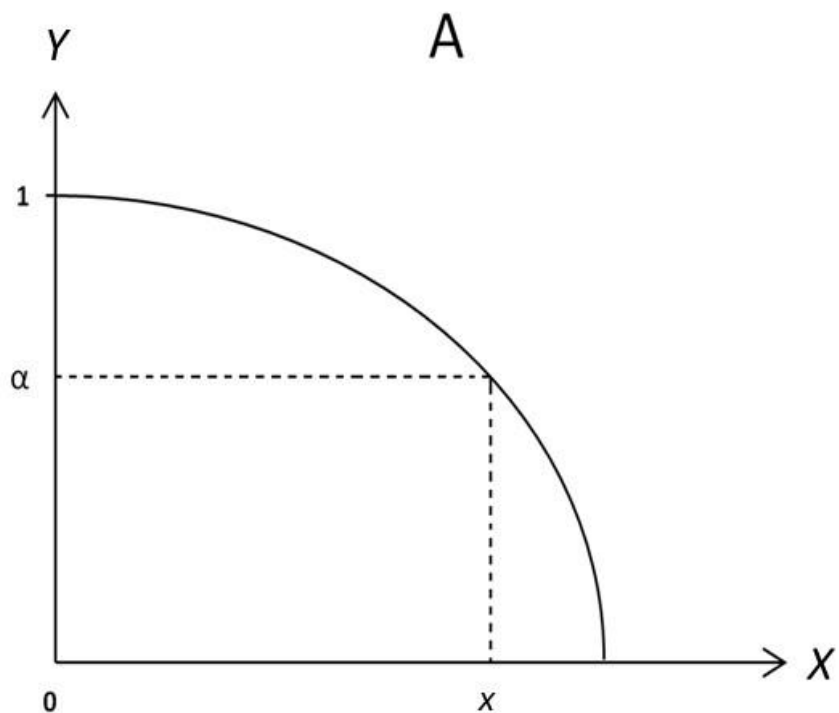
Un conjunto “Difuso/borroso” (*Fuzzy*) A posee una función de pertenencia definida en el conjunto universo $U = \mathbb{R}^n$ (por ejemplo) y está caracterizado por una función de inclusión o pertenencia $A(x)$ que toma valores en el conjunto $[0,1]$, es decir:

$$A(x): U \rightarrow [0,1]$$

En donde $A(x)$ representa el grado en el que $x \in U$ pertenece al conjunto difuso A .

La función de pertenencia es la representación de un conjunto difuso, de este modo es un elemento de estudio inmediato en la teoría difusa¹⁷. En la Figura 1 se puede observar la gráfica de la función de pertenencia para un conjunto difuso, de este ejemplo se puede observar que un valor “a” el cual pertenece a un conjunto difuso definido en U tiene asociado a él un grado de pertenencia (en este caso α) el cual está contenido en el conjunto $[0,1]$.

Figura 1. Representación de un conjunto difuso



7.2.1. Función de pertenencia y cortes de un conjunto difuso

La función de pertenencia además de ser la representación de un conjunto difuso, contiene todas las características que este conjunto posee. A continuación se

¹⁷ KLIR, George J. y YUAN, Bo. Fuzzy sets and fuzzy logic: Theory and applications. Upper Saddle River: Prentice Hall Inc, 1995. p. 1-19.

presentan algunas de las características del conjunto difuso a partir de su función de pertenencia.

Soporte de un conjunto difuso ${}^{0+}A$: Sea A un conjunto difuso contenido en un universo U y $A(x)$ su respectiva función de pertenencia, se define el soporte de A ($suppA$) como un conjunto en el sentido clásico conformado por todos los x para los cuales el (grado de pertenencia) nivel de pertenencia es mayor de cero y se denota del siguiente modo:

$${}^{0+}A = [x \in U | A(x) > 0]$$

Centro de un conjunto difuso 1A : Sea A un conjunto difuso *normal* cuya función de pertenencia es $A(x)$; se dice que el centro (*core*) de un conjunto difuso, considerado un conjunto clásico el cual se encuentra conformado por los valores de x que poseen el grado 1 de pertenencia al conjunto difuso. El *core* de un conjunto difuso es comúnmente denotado del siguiente modo:

$${}^1A = [A(x) = 1]$$

La altura de un conjunto difuso $h(A)$: Sea A un conjunto difuso contenido en un universo U y $A(x)$ su respectiva función de pertenencia, se considera la altura del conjunto difuso al conjunto *crisp* conformado por los valores de x para los cuales $A(x)$ toman el mayor grado de pertenencia dentro de dicho conjunto¹⁸.

7.2.2. Tipos de conjuntos difusos

A continuación se muestran algunos tipos de conjuntos difusos, de acuerdo con Klir y Yuan¹⁹, relevantes para el presente trabajo.

¹⁸ *Ibíd.*, p. 19-20.

¹⁹ *Ibíd.*, p. 20-24.

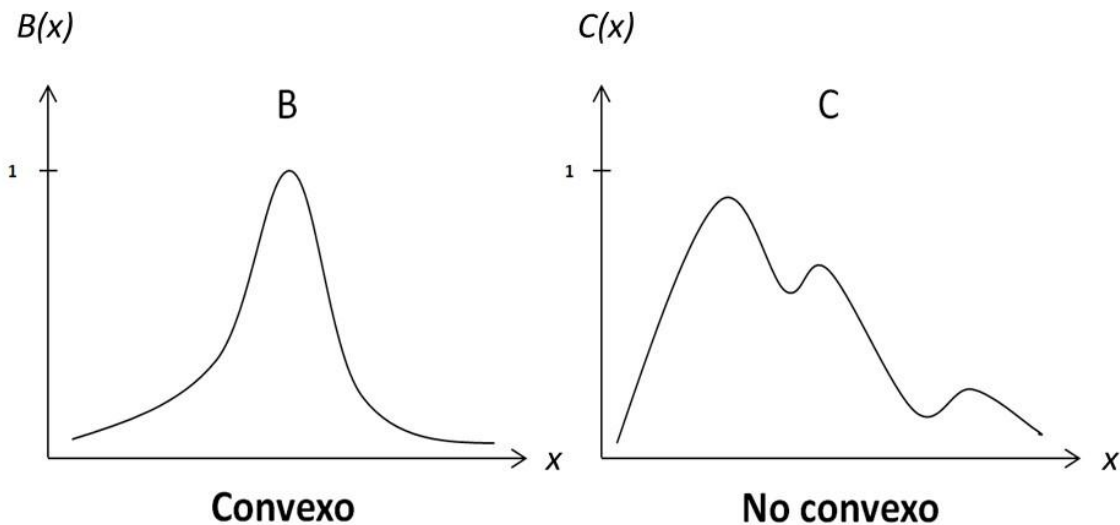
- *Conjunto difuso normal:* Un conjunto difuso es normal si y solo si existe por lo menos un $x \in A$ tal que $A(x) = 1$.
- *Conjunto difuso simétrico:* Un conjunto difuso es simétrico si y solo si existen dos valores $x_1, x_2 \in A$ equidistantes al centro (*core*) tal que $A(x_1) = A(x_2)$.
- *Conjunto difuso convexo:* Un conjunto difuso definido en los reales se denomina convexo si y solo si cumple con la siguiente restricción:

$$A((\lambda x_1) + (1 - \lambda)x_2) \geq \min[A(x_1), A(x_2)]$$

Para todo $x \in A, y \lambda \in [0,1]$

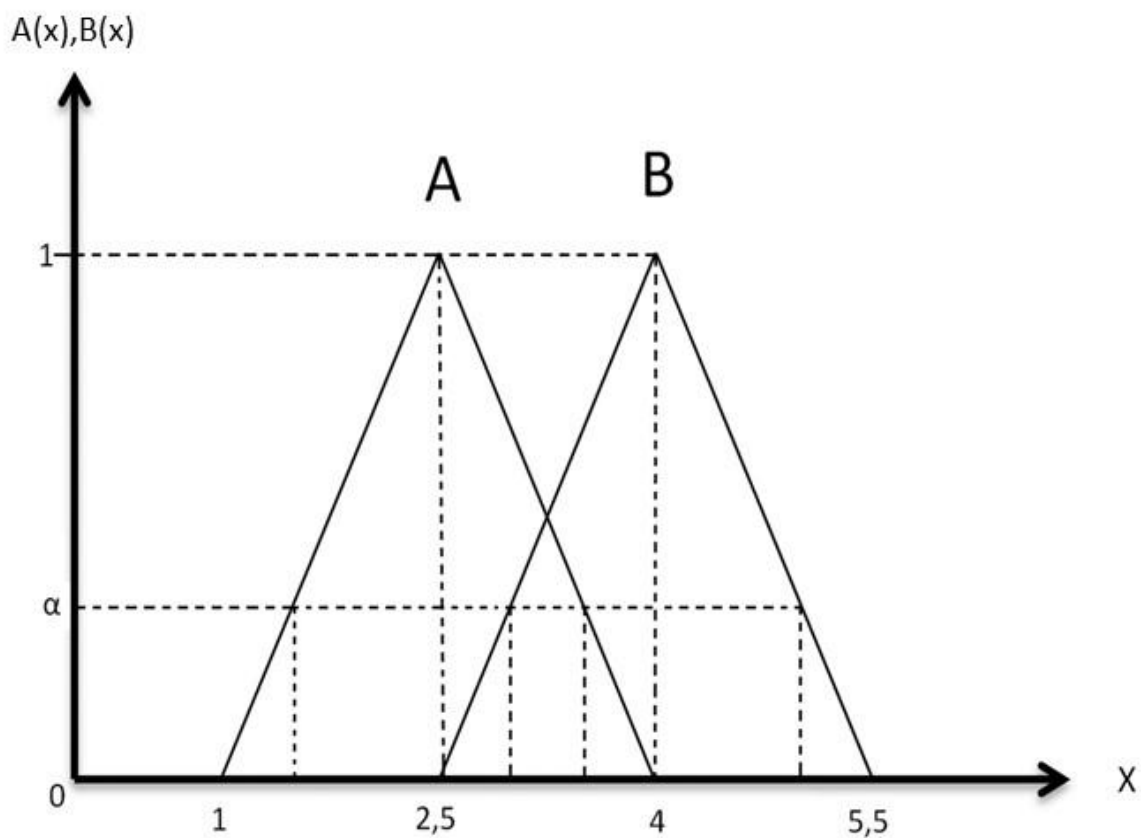
Un ejemplo de esto se puede ver en la Figura 2.

Figura 2. Ejemplo de un conjunto difuso convexo y uno no convexo



- *Conjunto difuso triangular*: Es todo aquel que posee una función de pertenencia conformada principalmente por dos funciones, una monótona creciente y una monótona decreciente, las cuales se interceptan entre sí en un punto del universo en discurso²⁰ U . Un ejemplo de un conjunto difuso triangular se observa en la Figura 3.

Figura 3. Ejemplo de dos conjuntos difusos triangulares



7.2.3. Corte α y corte estricto α

El concepto de los cortes alfa α en la matemática difusa es uno de los más importantes debido a la conexión que existe entre la matemática clásica y la difusa mediante estos últimos. Por otra parte, los α -cut y *strong* α -cut son otra manera de

²⁰ CANO. Op cit., p. 13-16.

representar un conjunto difuso debido a que estos son conjuntos clásicos que contienen todos los valores x que pertenecen al conjunto difuso y poseen niveles o grados de pertenencia superiores al valor α .

Conjunto α -cut:

$${}^{\alpha}A = [x \mid A(x) \geq \alpha]$$

Conjunto strong α -cut:

$${}^{\alpha+}A = [x \mid A(x) > \alpha]$$

Conjunto nivel: Sea A un conjunto difuso, y ${}^{\alpha}A$ un conjunto α -cut, se llama conjunto nivel de “ A ” al conjunto formado por todos los niveles α que pertenecen al intervalo $[0,1]$, que representan distintos α -cuts y se denota del siguiente modo:

$$\Delta(A) = [\alpha \mid A(x) = \alpha \text{ para algún } x \in X]$$

El siguiente ejemplo permite comprender las definiciones expuestas anteriormente. Sean $U = \mathbb{R}$ y los conjuntos difusos A y B , representados mediante sus funciones de pertenencia:

$$A(x) = \begin{cases} \frac{2x}{3} - \frac{2}{3} & \text{para } 1 < x < 2.5 \\ -\frac{2x}{3} + \frac{8}{3} & \text{para } 2.5 < x < 4 \\ 0 & \text{para } 1 \geq x \geq 4 \end{cases}$$

$$B(x) = \begin{cases} \frac{2x}{3} - \frac{5}{3} & \text{para } 2.5 < x < 4 \\ -\frac{2x}{3} + \frac{11}{3} & \text{para } 4 < x < 5.5 \\ 0 & \text{para } 2.5 \geq x \geq 5.5 \end{cases}$$

Los conjuntos α -cut y strong α -cut para cada conjunto difuso serían los siguientes²¹:

$$\begin{aligned} \alpha A &= (1 + 1.5\alpha, 4 - 1.5\alpha) \\ \alpha^+ A &= [1 + 1.5\alpha, 4 - 1.5\alpha] \\ \alpha B &= (2.5 + 1.5\alpha, 5.5 - 1.5\alpha) \\ \alpha^+ B &= [2.5 + 1.5\alpha, 5.5 - 1.5\alpha] \end{aligned}$$

7.2.4. Operaciones básicas entre conjuntos difusos

Las tres operaciones básicas en la teoría clásica de conjuntos, “complemento estándar, unión e intersección” se trasladan a la teoría difusa y es necesario observar de forma general cada una de estas operaciones²².

Sea \bar{A} el complemento estándar (véase Figura 4) de un conjunto difuso A con respecto al universo X . $\bar{A}(x)$ se define para cada $x \in X$ como:

$$\bar{A}(x) = 1 - A(x)$$

Sean A y B dos conjuntos difusos, se tiene que la unión estándar (véase Figura 5) e intersección estándar (véase Figura 6), $(A \cup B)$ y $(A \cap B)$ están dados para cada $x \in X$ de la siguiente manera:

²¹ KLIR y YUAN. Ob cit., p. 19-20.

²² Ibíd., p. 25-27.

$$(A \cup B)(x) = \max[A(x), B(x)]$$

$$(A \cap B)(x) = \min[A(x), B(x)]$$

Figura 4. Complemento estándar entre conjuntos difusos

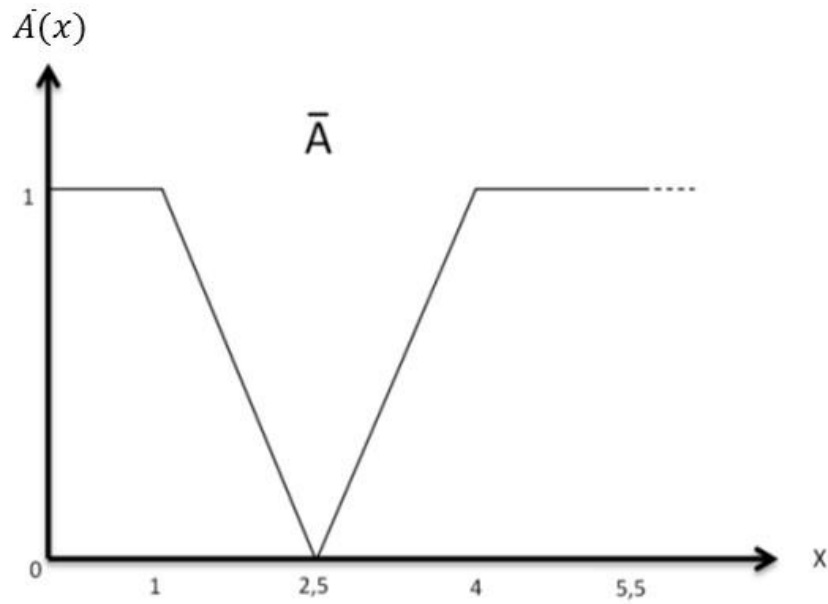


Figura 5. Unión estándar de conjuntos difusos

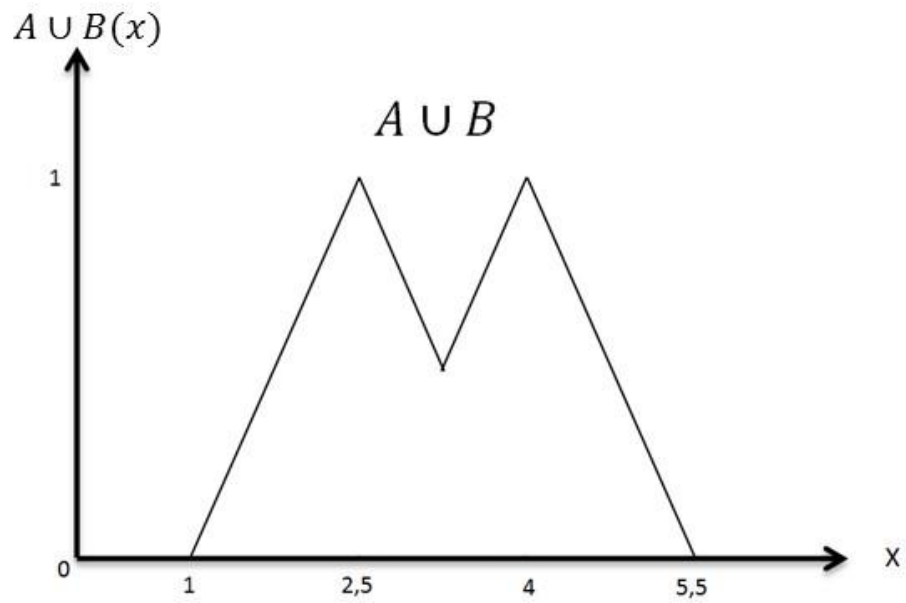
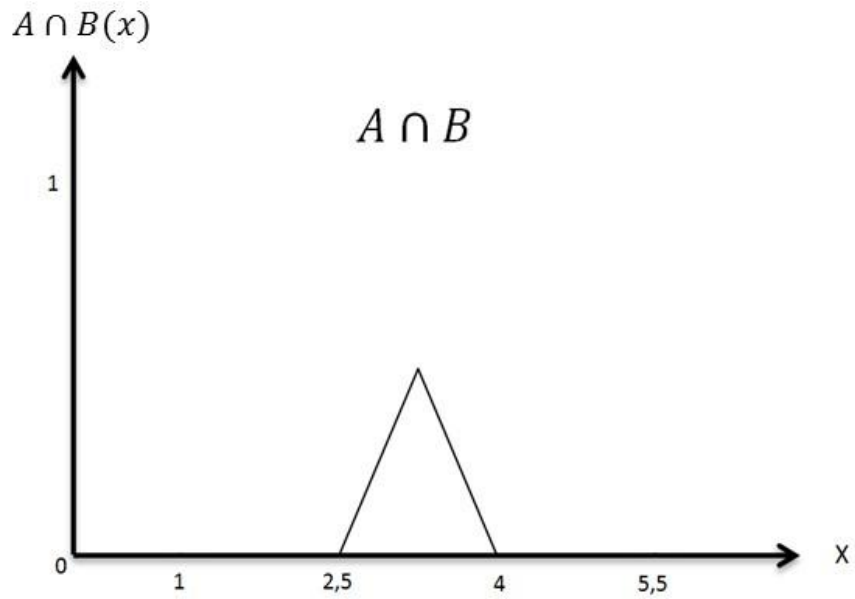


Figura 6. Intercepción estándar entre conjuntos difusos



7.2.5. Notación para conjuntos difusos discretos o finitos

Sea A un conjunto difuso definido en un universo X , sean x_1, x_2, \dots, x_n los elementos de $SuppA$ y sea a_i los grados de pertenencia de x_i , entonces A se describe del siguiente modo:

$$A(x) = a_1/x_1 + a_2/x_2 + \dots \dots \dots a_r/x_n$$

Otras notaciones utiles se ven a continuacion:

$$A = \sum_{i=1}^{\infty} a_i/x_i$$

$$A = \int_x^{\infty} A(x)/x$$

En donde a_i es el grado de pertenencia asociado al valor de x_i respectivamente en el conjunto difuso A ²³.

7.2.6. Representación de conjuntos difusos

La representación de un conjunto difuso se realiza mediante los conjuntos α -cut y *strong* α -cut. Muchas de las propiedades y operaciones de la teoría clásica de conjuntos tienen su contraparte difusa, dichas extensiones son llamadas cada una *cutworthy* y *strong cutworthy* dependiendo del tipo de representación empleada²⁴.

Sea un conjunto difuso finito A , se discute ahora la manera de representar el conjunto por una familia de α -cuts:

$$A = 0.1/x_1 + 0.3/x_2 + 1/x_3$$

α -cuts de A :

$${}^{0.1}A = 1/x_1 + 1/x_2 + 1/x_3$$

$${}^{0.3}A = 0/x_1 + 1/x_2 + 1/x_3$$

$${}^1A = 0/x_1 + 0/x_2 + 1/x_3$$

Ahora, se define un nuevo conjunto difuso ${}_{\alpha}A$ a partir de los α -cuts de A siguiendo el procedimiento que se observa a continuación:

$${}_{\alpha}A = \alpha \cdot {}^{\alpha}A(x)$$

$${}_{0.1}A = 0.1/x_1 + 0.1/x_2 + 0.1/x_3$$

$${}_{0.3}A = 0/x_1 + 0.3/x_2 + 0.3/x_3$$

²³ *Ibíd.*, p. 28-29.

²⁴ *Ibíd.*, p. 39-43.

$${}_1A = 0/x_1 + 0/x_2 + 1/x_3$$

De la unión de los ${}_\alpha A$ se obtiene nuevamente el Conjunto Difuso A .

Con lo anterior observado, se pueden discutir los siguientes tres teoremas de descomposición de un conjunto difuso, dado que son de importancia para el presente trabajo.

- *Primer teorema de descomposición:*

Para cada conjunto difuso A el cual pertenece a una función $f(x)$ definida en los reales; se tiene que:

$$A = \bigcup_{\alpha \in [0,1]} {}_\alpha A$$

- *Segundo teorema de descomposición:*

Para cada conjunto difuso A el cual pertenece a una función $f(x)$ definida en los reales; se tiene que:

$$A = \bigcup_{\alpha \in [0,1]} {}_{\alpha+} A$$

- *Tercer teorema de descomposición:*

Para cada conjunto difuso A el cual pertenece a una función $f(x)$ definida en los reales; se tiene que:

$$A = \bigcup_{\alpha \in \Lambda(A)} {}_\alpha A$$

7.2.7. Principio de extensión para los conjuntos difusos

Cuando se habla del principio de extensión de la matemática difusa, se refiere a la *fuzzificación* de las funciones $f(x)$, que se da mediante su interacción con los conjuntos difusos definidos también en X y Y ²⁵.

El teorema del principio de extensión dice:

Se tiene una clásica función de la forma $F: X \rightarrow Y$, y su inversa $F^{-1}: Y \rightarrow X$,

$$F: X \rightarrow Y$$

$$F^{-1}: Y \rightarrow X$$

Entonces, para todo $A \in \mathcal{F}(X)$ se tiene que,

$$[F(A)](y) = \sup_{x|y=F(x)} A(x)$$

Para todo $B \in \mathcal{F}(Y)$ se tiene que,

$$[F^{-1}(B)](x) = B(F(x))$$

Se podría decir que una función $f(x)$ se fuzzifica cuando esta toma los valores de x que pertenecen a un conjunto difuso A definido en x , creando un conjunto difuso imagen en el eje y . Del mismo modo sucede para la función inversa.

7.2.8. Aritmética difusa

Dentro de la matemática difusa existen diferentes tipos de conjuntos difusos como se ha visto a través del texto, sin embargo aquellos que están definidos en los

²⁵ *Ibíd.*, p. 44-48.

reales pasan a tomar un mayor grado de importancia debido a que entre estos últimos se encuentran los que pueden ser llamados números difusos.

Se denota por $\mathcal{F}(\mathbb{R})$ el conjunto de todos los subconjuntos fuzzy A de \mathbb{R} con función de pertenencia $A(x)$ que satisface las condiciones:

1. A es un conjunto normal (ver sección 7.2.2).
2. La función $A(x)$ es cuasi-cóncava, esto es $A(\lambda x + (1 - \lambda)y) \geq \min(A(x), A(y))$.
3. $A(x)$ es semi-continua superiormente, significa que $\{x \in \mathbb{R}: A(x) \geq \alpha\}$. Esto es, los α -cuts de A son intervalos cerrados y acotados, por consiguiente son conjuntos compactos.
4. El soporte de A , ${}^{0+}A$ es acotado, por consiguiente es compacto.

Un elemento A en $\mathcal{F}(\mathbb{R})$ es un número difuso.

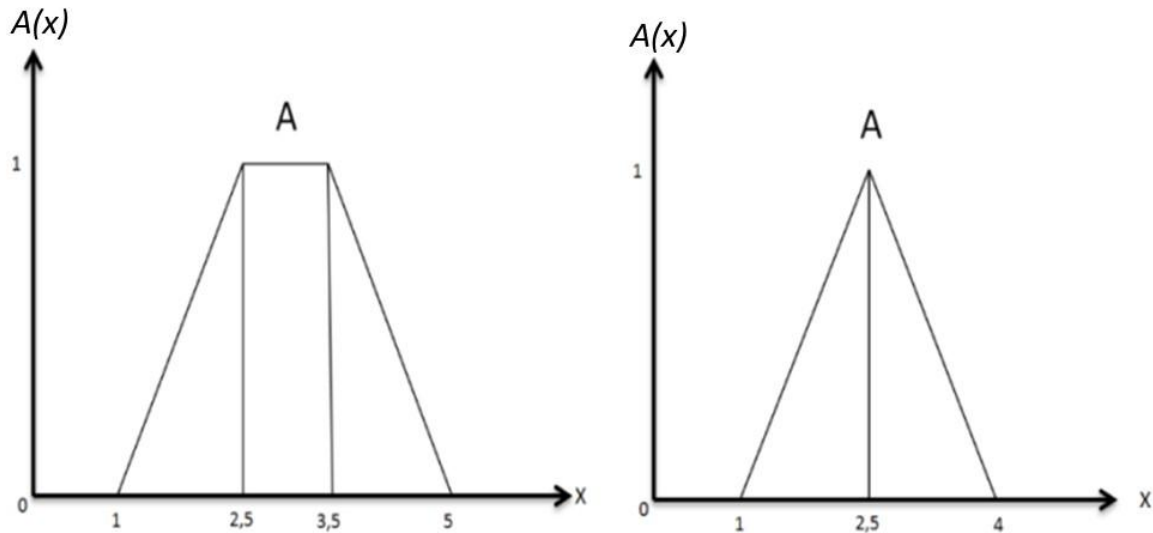
Los números difusos o intervalos difusos siempre deben de ser convexos pero solo los números difusos cumplen con el teorema de normalidad, es así como se diferencian entre sí, esto debido a que en un intervalo difuso existe más de un valor de $x \in \mathbb{R}$ para el cual su nivel de pertenencia sea 1.

Dentro de la Aritmética difusa se encuentra el Teorema de conformación de un número o intervalo difuso, el cual se define como: Sea A un conjunto difuso definido en los reales, A es un número difuso si y solo si existe un intervalo cerrado $[a, b]$ tal que:

$$A(x) = \begin{cases} 1 & \text{Para } x \in [a, b] \\ L(x) & \text{Para } x \in (-\infty, a) \\ R(x) & \text{Para } x \in (b, \infty) \end{cases}$$

En donde L es una función monótona creciente continua por la derecha y R es una función monótona decreciente continua por la izquierda²⁶.

Figura 7. Ejemplo de un número difuso y un intervalo difuso



7.2.9. Variable lingüística

Los números difusos son comúnmente usados para la representación de los estados de las variables lingüísticas cuantitativas, estos estados comúnmente son asignados a ciertos cambios en un entorno al cual se va a analizar y son representados mediante conceptos lingüísticos generando así una estructura denominada variable lingüística. (Ver ejemplo en la Figura 8).

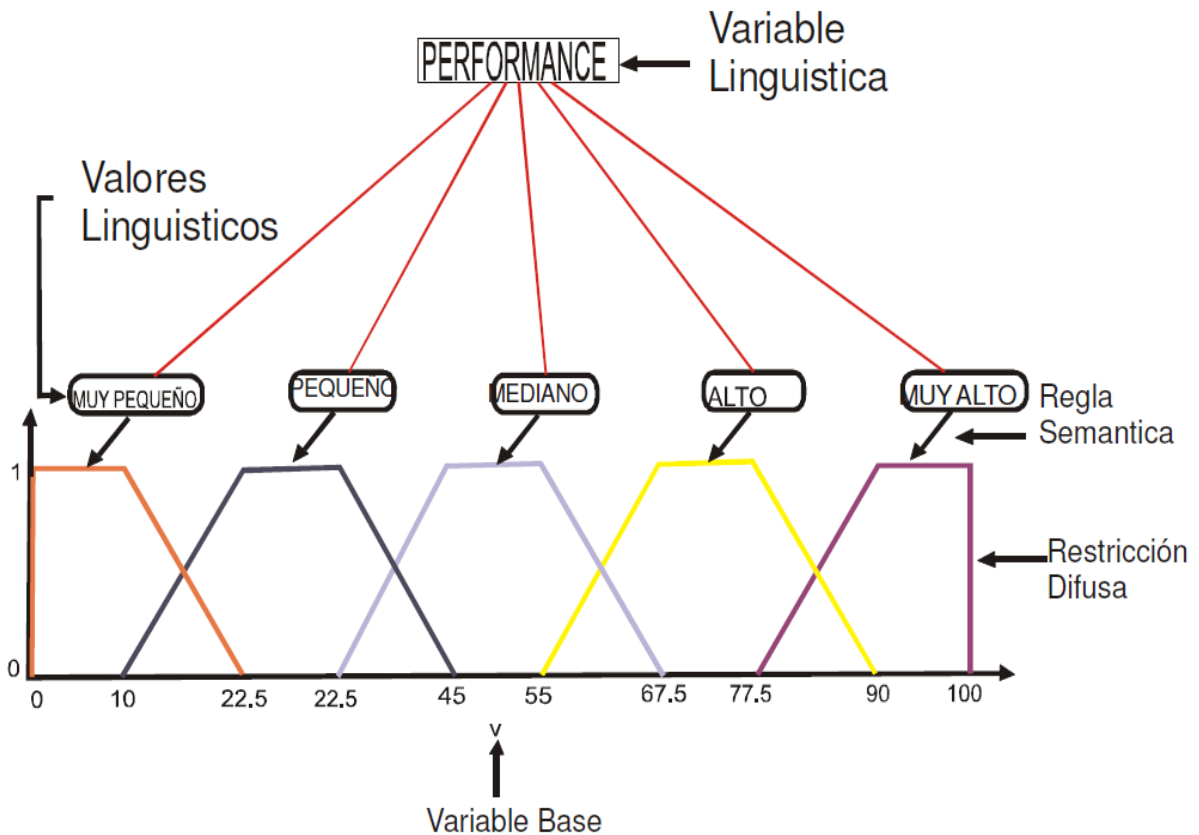
Una variable fuzzy se describe mediante la quintupla (v, T, X, g, m) en donde:

- v es el nombre de la variable lingüística.
- X es el conjunto universo.

²⁶ *Ibíd.*, p. 39-43.

- T es el conjunto de términos lingüísticos de v en una variable base con valores que se encuentran en el conjunto X .
- g es una regla sintáctica (una gramática) por lo general los términos lingüísticos en T .
- m es una regla semántica que asigna a cada término lingüístico $t \in T$ su significado $m(t)$ que es un conjunto difuso en X . Es decir $m: T \rightarrow \mathcal{F}(X)$.²⁷

Figura 8. Ejemplo de una variable lingüística



Fuente: George J. Klir y Bo Yuan²⁸

²⁷ Ibíd., p. 102.

²⁸ Ibíd., p. 101.

7.2.10. Operaciones aritméticas en intervalos difusos

La aritmética difusa se basa en dos propiedades que fueron mencionadas con anterioridad, siendo las siguientes:

- Cada conjunto difuso puede ser completamente representado por su conjunto de α -cuts.
- Los α -cuts de cada conjunto difuso son intervalos cerrados de números reales para cada $\alpha \in (0,1]$.

Estas propiedades son la base fundamental para las operaciones aritméticas entre números difusos, operaciones que se basan en términos de sus α -cuts.

Es importante definir la suma, resta, multiplicación, división de números fuzzy. Como se señaló toda esta aritmética se basa en sus α -cuts, esto es, sobre los intervalos cerrados.

Sea $*$ cualquier operación aritmética básica (+ suma, - sustracción, x multiplicación, / división), entonces:

$$[a, b] * [d, e] = \{f * g \mid a \leq f \leq b, d \leq g \leq e\}$$

No incluye $[a, b]/[d, e]$ cuando $0 \in [d, e]$

Esta ecuación es una forma general de ver cualquier operación aritmética entre intervalos cerrados. En la Tabla 1 se encuentran algunos ejemplos para dar una mayor comprensión sobre las definiciones dadas.

Estas operaciones entre intervalos cerrados satisfacen ciertas propiedades como las que se encuentran en la Tabla 2²⁹:

Tabla 1. Ejemplos de operaciones aritméticas

Suma	$[a, b] + [d, e] = [a + d, b + e]$
Resta	$[a, b] - [d, e] = [a - e, b - d]$
Multiplicación	$[a, b] \cdot [d, e] = [\min(ad, ae, bd, be), \max(ad, ae, bd, be)]$
División	$\frac{[a, b]}{[d, e]} = \left[\min\left(\frac{a}{d}, \frac{a}{e}, \frac{b}{d}, \frac{b}{e}\right), \max\left(\frac{a}{d}, \frac{a}{e}, \frac{b}{d}, \frac{b}{e}\right) \right]$ <p style="text-align: center;">Cuando $0 \notin [d, e]$</p>

Tabla 2. Propiedades entre intervalos cerrados

Conmutativa	$A + B = B + A$ $A \cdot B = B \cdot A$
Asociativa	$(A + B) + C = A + (B + C)$ $(A \cdot B) \cdot C = A \cdot (B \cdot C)$
Identidad	$A = 0 + A = A + 0$ $A = 1 \cdot A = A \cdot 1$
Subdistributiva	$A \cdot (B + C) \subseteq A \cdot B + A \cdot C$

²⁹ *Ibíd.*, p. 102-104.

7.2.11. Operaciones aritméticas entre números difusos

A continuación se exponen dos métodos para la resolver las operaciones básicas entre números difusos, el primero de los métodos se basa en la aritmética de intervalos y el segundo método empleado es el principio de extensión. Es necesario asumir que los números difusos que se discuten se representan mediante funciones de pertenencia continuas.

Sean A y B dos conjuntos difusos definidos en los \mathbb{R} , y $*$ cualquier operación aritmética básica. Ahora, se definen los α -cuts ${}^\alpha(A * B)$ de la siguiente forma:

$${}^\alpha(A * B) = {}^\alpha A * {}^\alpha B$$

Para todo $\alpha \in (0,1]$. Si $*$ se requiere que $0 \notin {}^\alpha B$ para todo $\alpha \in (0,1]$.

En base al primer teorema de descomposición, $A * B$ puede ser expresado de la siguiente forma:

$$A * B = \bigcup_{\alpha \in [0,1]} {}^\alpha(A * B)$$

Dado que A y B son números difusos definidos en los reales, entonces $A * B$ también es un número difuso.

Ejemplo: Sean los dos conjunto difusos triangulares A y B con las siguientes funciones de pertenencia y α -cuts.

$$A(x) = \begin{cases} 0 & \text{para } x \leq -1 \text{ y } x > 3 \\ \frac{x+1}{2} & \text{para } -1 < x \leq 1 \\ \frac{3-x}{2} & \text{para } 1 < x \leq 3 \end{cases}$$

$$B(x) = \begin{cases} 0 & \text{para } x \leq 1, x > 5 \\ \frac{x-1}{2} & \text{para } 1 < x \leq 3 \\ \frac{5-x}{2} & \text{para } 3 < x \leq 5 \end{cases}$$

$${}^{\alpha}A = (2\alpha - 1, 3 - 2\alpha)$$

$${}^{\alpha}B = (2\alpha + 1, 5 - 2\alpha)$$

Aplicando los conceptos de aritmética de intervalos se tiene,

$${}^{\alpha}(A + B) = [4\alpha, 8 - 4\alpha] \text{ para } \alpha \in (0,1]$$

$${}^{\alpha}(A - B) = [4\alpha - 6, 2 - 4\alpha] \text{ para } \alpha \in (0,1]$$

$${}^{\alpha}(A \cdot B) = \begin{cases} [-4\alpha^2 + 12\alpha - 5, 4\alpha^2 - 16\alpha + 15] & \text{para } \alpha \in (0, .5] \\ [4\alpha^2 - 1, 4\alpha^2 - 16\alpha + 15] & \text{para } \alpha \in (.5, 1] \end{cases}$$

$${}^{\alpha}(A/B) = \begin{cases} \left[\frac{2\alpha - 1}{2\alpha + 1}, \frac{3 - 2\alpha}{2\alpha + 1} \right] & \text{para } \alpha \in (0, .5] \\ \left[\frac{2\alpha - 1}{5 - 2\alpha}, \frac{3 - 2\alpha}{2\alpha + 1} \right] & \text{para } \alpha \in (.5, 1] \end{cases}$$

Como se discutió, los resultados de las operaciones básicas entre dos números difusos son también un número difuso, como un ejemplo se puede ver el resultado de la suma y la resta por medio de sus funciones de pertenencia:

$$(A + B)(x) = \begin{cases} 0 & \text{para } x \leq 0, x > 8 \\ \frac{x}{4} & \text{para } 0 < x \leq 4 \\ \frac{8-x}{2} & \text{para } 4 < x \leq 8 \end{cases}$$

$$(A - B)(x) = \begin{cases} 0 & \text{para } x \leq -6, x > 2 \\ \frac{x + 6}{4} & \text{para } -6 < x \leq 3 \\ \frac{2 - x}{2} & \text{para } 3 < x \leq 5 \end{cases}$$

El segundo método para resolver operaciones aritméticas básicas entre números difusos, es el mediante el principio de extensión visto ya con anterioridad.

Sean A y B dos números difusos definidos en los reales y $(A * B)$ otro número difuso, siendo este último el resultado de una operación aritmética básica cualesquiera simbolizada como $*$. Entonces se tiene que:

$$(A * B)(z) = \sup_{z=x*y} \min[A(x), B(y)]$$

Para todo $z \in \mathbb{R}$

De este modo, cada una de las operaciones básicas sería vista de la siguiente forma³⁰:

$$(A + B)(z) = \sup_{z=x+y} \min[A(x), B(y)]$$

$$(A - B)(z) = \sup_{z=x-y} \min[A(x), B(y)]$$

$$(A \cdot B)(z) = \sup_{z=x \cdot y} \min[A(x), B(y)]$$

$$(A/B)(z) = \sup_{z=x/y} \min[A(x), B(y)]$$

³⁰ *Ibíd.*, p. 105-109.

7.3. PROGRAMACIÓN LINEAL DIFUSA (FLP)

La programación lineal es una rama de la ingeniería enfocada a la construcción de algoritmos matemáticos usados como fundamento de una decisión empresarial. Tiene como objetivo encontrar un valor lo más cercano al óptimo posible (mínimo o máximo) de una función lineal denominada “*Función objetivo*” la cual, se encuentra delimitada por un grupo de restricciones (funciones lineales).

Un problema de programación lineal se define de la siguiente manera:

$$\begin{aligned} F.O. \quad & \text{Min (or Max)} \quad z = cx \\ \text{s. a.} \quad & Ax \leq b \\ & x \geq 0 \end{aligned}$$

- En donde $x = (x_1, x_2, \dots, \dots, x_n)^T$ es un vector de variables, denominadas variables de decisión.
- El conjunto de vectores x que satisfacen todas las restricciones es llamado conjunto factible.

La Programación Lineal Difusa - FLP (por sus siglas en inglés *Fuzzy Linear Programming*) a su vez, se define y tiene el mismo objetivo que la programación lineal tradicional, sin embargo, los modelos matemáticos construidos en este caso en particular, poseen algún tipo de parámetro difuso, con lo cual, es necesario profundizar en el tema³¹.

7.3.1. Tipos de programación lineal difusa

Un problema de programación lineal difusa se formula de la siguiente forma:

³¹ *Ibíd.*, p. 408-409.

$$\begin{aligned}
 & \text{Max } \sum_{j=1}^n C_j X_j \\
 \text{S. A. } & \sum_{j=1}^n A_{ij} X_j \leq B_i \text{ para todo } (i \in \mathbb{N}_m) \\
 & X_j \geq 0 \text{ para todo } (j \in \mathbb{N}_m)
 \end{aligned}$$

En donde A_{ij} , B_i y C_j son números difusos y X_j son variables cuyos estados son números difusos. Las operaciones de suma y multiplicación se dan a partir de aritmética entre números difusos.

No necesariamente todos los parámetros del modelo expuesto anteriormente son números difusos, a continuación se discuten dos casos particulares en que algunos parámetros son “*crisp*” y otros difusos:

- Caso 1: Solo el termino independiente (lado derecho de las restricciones) “ B_i ” son números difusos. Dicho problema se describe de la forma,

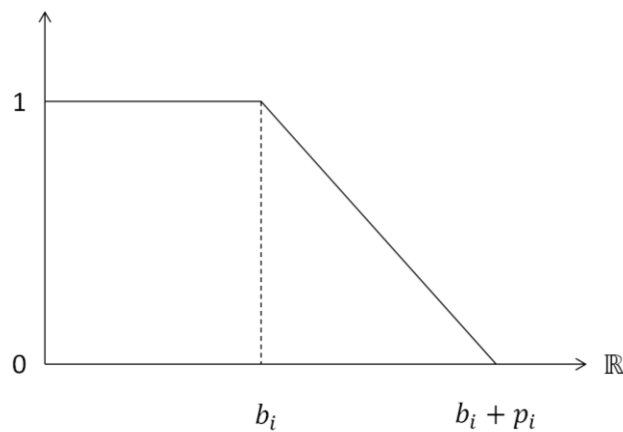
$$\begin{aligned}
 & \text{Max } \sum_{j=1}^n c_j x_j \\
 \text{S. A. } & \sum_{j=1}^n a_{ij} x_j \leq B_i \text{ para todo } (i \in \mathbb{N}_m) \\
 & x_j \geq 0 \text{ para todo } (j \in \mathbb{N}_m)
 \end{aligned}$$

En la Figura 9 se muestra un caso particular del número fuzzy B_i , con función de pertenencia:

$$B_i(x) = \begin{cases} 1 & \text{cuando } x_i \leq b_i \\ \frac{b_i + p_i - x}{p_i} & \text{cuando } b_i < x < b_i + p_i \\ 0 & \text{cuando } b_i + p_i \leq x \end{cases}$$

En donde $x \in \mathbb{R}$, para cada vector $x = (x_1, x_2, \dots, x_n)$

Figura 9. Número difuso en el caso 1



Fuente: Klir y Yuan³²

Ahora, se procede a hallar el grado $D_i(x)$ con que x satisface la i th restricción ($i \in \mathbb{N}_m$) del siguiente modo:

$$D_i(x) = B_i\left(\sum_{j=0}^n a_{ij}x_j\right)$$

Estos grados de un conjunto difuso en \mathbb{R}^n , y su intersección $\bigcap_{i=1}^m D_i$, son un conjunto difuso factible.

El conjunto difuso de valores óptimos del problema de programación lineal difusa es un conjunto difuso, que se determina mediante los límites “superior e inferior”.

³² *Ibíd.*, p. 412

El límite inferior:

$$\max z_l = cx$$

$$S.A. \sum_{j=1}^n a_{ij} x_j \leq b_i \text{ para todo } (i \in \mathbb{N}_m)$$

$$x_j \geq 0 \text{ para todo } (j \in \mathbb{N}_m);$$

El límite superior:

$$\max z_u = cx$$

$$S.A. \sum_{j=1}^n a_{ij} x_j \leq b_i + p_i \text{ para todo } (i \in \mathbb{N}_m)$$

$$x_j \geq 0 \text{ para todo } (j \in \mathbb{N}_m);$$

Luego, se procede a encontrar el subconjunto "G",

$$G(x) = \begin{cases} 1 & \text{cuando } z_u \leq cx \\ \frac{cx - z_l}{z_u - z_l} & \text{cuando } z_l \leq cx \leq z_u \\ 0 & \text{cuando } cx \leq z_l \end{cases}$$

De este modo el caso 1, se transforma en el siguiente problema de optimización clásica:

$$\max \lambda$$

$$S.A. \lambda(z_u - z_l) - cx \leq -z_l$$

$$\lambda p_i + \sum_{i=1}^n a_{ij} x_j \leq b_i + p_i \quad (i \in \mathbb{N}_m)$$

$$\lambda, x_j \geq 0 \quad (j \in \mathbb{N}_n)$$

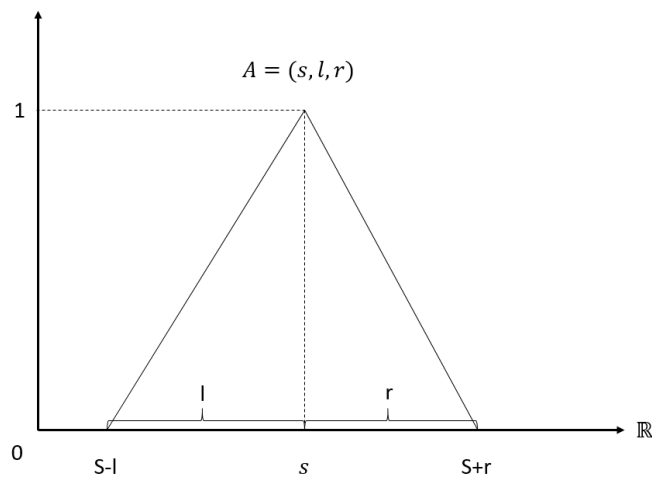
- Caso 2: El lado derecho “ B_i ” y los coeficientes “ A_{ij} ” de la matriz restricción son números difusos.

$$\begin{aligned} & \text{Max} \sum_{j=1}^n c_j x_j \\ \text{S. A.} \quad & \sum_{j=1}^n A_{ij} x_j \leq B_i \text{ para todo } (i \in \mathbb{N}_m) \end{aligned}$$

$$x_j \geq 0 \text{ para todo } (j \in \mathbb{N}_m)$$

Para la explicación de este caso se asumirá que todos los números difusos serán triangulares. De este modo, cada número difuso será representado por tres números reales $(s-l, s, s+r)$, los cuales se observan en la Figura 10.

Figura 10. Número difuso triangular para el caso 2



Fuente: Klir y Yuan³³

³³ *Ibíd.*, p. 412

El problema para el caso dos puede ser formulado del siguiente modo para un número difuso $A = (s, l, r)$:

$$\begin{aligned} & \text{Max o Min } \sum_{j=1}^n c_j x_j \\ \text{S.A. } & \sum_{j=1}^n (s_{ij}, l_{ij}, r_{ij}) x_{ij} \leq (t_i, u_i, v_i) \quad (i \in \mathbb{N}_m) \\ & x_j \geq 0 \quad (j \in \mathbb{N}_n) \end{aligned}$$

Claramente se puede ver que $A_{ij} = (s_{ij}, l_{ij}, r_{ij})$ y $B_i = (t_i, u_i, v_i)$ son cada uno números difusos. De este modo el problema quedaría escrito del siguiente modo³⁴:

$$\begin{aligned} & \text{Max o Min } \sum_{j=1}^n c_j x_j \\ \text{S.A. } & \sum_{j=1}^n s_{ij} x_j \leq t_i \\ & \sum_{j=1}^n (s_{ij} - l_{ij}) x_j \leq t_i - u_i \\ & \sum_{j=1}^n (s_{ij} + r_{ij}) x_j \leq t_i + v_i \quad (i \in \mathbb{N}_m) \\ & x_j \geq 0 \quad (j \in \mathbb{N}_n) \end{aligned}$$

³⁴ *Ibíd.*, p. 409-415.

8. FORMULACIÓN DEL PROBLEMA CVRP CON DEMANDA DIFUSA

8.1. CVRP vs FVRP

El CVRP (*Capacited Vehicle Routing Problem*) como se dijo con anterioridad, es un modelo de programación lineal básico, cuyo objetivo es dar solución a los problemas de ruteo de vehículos con capacidad fija CI . A continuación, se expondrá el modelo en su forma clásica, para posteriormente modificarlo a su contraparte difusa.

El conjunto de índices i, j , representan a los clientes en donde, 0 (cero) denota el depósito. Sea, C_{ij} la distancia recorrida de un vehículo k , desde i hasta j . n es la longitud de i y j respectivamente y p la longitud de k .

Las variables de decisión se denotan como X_{ijk} y Y_{ik} , en donde, X toma el valor de 1 si el vehículo k realiza el recorrido de i a j o 0 (cero) de lo contrario. Por otro lado, Y toma el valor de 1 si el vehículo k sirve al cliente i de lo contrario será 0 (cero).

La distancia máxima recorrida por cada vehículo k será L y d_i será la demanda del cliente ubicado en el lugar i .

La función objetivo es la suma de las distancias recorridas por cada vehículo y se expresa de la siguiente forma:

$$F.O.$$
$$Min z = \sum_i^n \sum_j^n \sum_k^p C_{ij} X_{ijk}$$

Las siguientes dos restricciones expresan la relación entre las variables x y y :

s. a.

$$\sum_i^n X_{ijk} = Y_{jk} \quad j = 0,1,2,\dots,n; k = 1,2,\dots,p.$$

$$\sum_j^n X_{ijk} = Y_{ik} \quad i = 0,1,2,\dots,n; k = 1,2,\dots,p.$$

Esta restricción evita que un vehículo realice el mismo recorrido más de una vez:

$$\sum_i^n X_{ijk} = \sum_i^n X_{jik} \quad j = 0,1,2,\dots,n; k = 1,2,\dots,p.$$

La siguiente restricción crea los conjuntos de clientes que pueden ser atendidos en un solo recorrido por el vehículo k :

$$\sum_i^n d_i Y_{ik} \leq CI \quad k = 1,2,\dots,p.$$

La restricción presentada a continuación garantiza que cada cliente sea visitado una vez:

$$\sum_i^n \sum_k^p X_{ijk} = 1 \quad j = 0,1,2,\dots,n.$$

Restricción para que cada vehículo salga y regrese a la bodega:

$$\sum_j^n X_{0jk} = 1 \quad k = 1, 2, \dots, p.$$

Restricción que determina la máxima distancia recorrida posible para cada vehículo:

$$\sum_i^n \sum_j^n C_{ij} X_{ijk} \leq L \quad k = 1, 2, \dots, p.$$

$$X, Y \in [0, 1], i, j = 0, 1, 2, \dots, n, k = 1, 2, \dots, p.$$

El problema consiste en hallar las rutas vehiculares que minimicen el valor de la función objetivo. En el problema se considera que todos los parámetros que figuran en el problema se conocen con certeza, lo cual se traduce en que los parámetros son determinísticos.

En el presente trabajo se considera la demanda como un número *fuzzy*. Por consiguiente, todo lo discutido anteriormente respecto a la matemática difusa se usa en lo que respecta al cumplimiento de las restricciones, evaluaciones de las funciones, etcétera.

Por ejemplo, las soluciones factibles a este problema pueden obtenerse aplicando los conceptos vistos en la sección anterior acerca del tema de resolución de problemas de programación lineal Fuzzy, en donde la restricción que crea los conjuntos de clientes se puede escribir del siguiente modo:

Se asume que la demanda es un número difuso triangular, esto es:

Sea d_i la demanda del cliente i , esta se puede escribir como (d_{i1}, d_{i2}, d_{i3}) entonces la restricción se vería de la siguiente manera:

$$\sum_i^n (d_{i1}, d_{i2}, d_{i3}) Y_{ik} \leq CI \quad k = 1, 2, \dots, p.$$

La manera tradicional de dar solución a este problema es simplemente usar el valor máximo dentro del conjunto difuso d_i viéndose finalmente de la siguiente manera:

$$\sum_i^n \text{Max}(d_{i1}, d_{i2}, d_{i3}) Y_{ik} \leq CI \quad k = 1, 2, \dots, p.$$

De esta forma se maneja el problema en su forma clásica eliminando el parámetro difuso, aun así, esta manera no sería recomendable, debido a que no siempre los clientes piden el máximo valor y por tanto las rutas generadas por este método serían las rutas 100% seguras y con esto los costos de transporte serían elevados. Según los conceptos introducidos por Zheng y Liu³⁵ en el tema de los números difusos, existe un método para dar manejo a este parámetro difuso mediante el concepto de credibilidad, de tal forma se pueda forzar a un vehículo a realizar una ruta sin tener completa certeza en poder cumplirla.

³⁵ ZHENG y LIU. Op cit., p. 673-683

8.2. CREDIBILIDAD Y SU APLICACIÓN AL FVRP

Zheng y Liu³⁶ afirman que los números difusos pueden ser representados por sus ecuaciones de posibilidad, necesidad y credibilidad. Para el caso en discusión, se hablaría de la credibilidad de poder atender la demanda de un cliente i una vez se hayan atendido n clientes en esa misma ruta y por consiguiente con ese mismo vehículo.

Sea V un número difuso triangular de la forma $V = (v_1, v_2, v_3)$, entonces, se tiene que su función de credibilidad para $v \geq q$ esta descrita del siguiente modo:

$$Cre\{v \geq q\} = \begin{cases} 1 & \text{si } q \leq v_1 \\ \frac{2v_2 - v_1 - q}{2(v_2 - v_1)} & \text{si } v_1 \leq q \leq v_2 \\ \frac{v_3 - q}{2(v_3 - v_2)} & \text{si } v_2 \leq q \leq v_3 \\ 0 & \text{si } q \geq v_3 \end{cases}$$

De este modo se entiende que la credibilidad $Cre \in [0,1]$.

Ahora, a continuación se discute la manera de determinar la capacidad residual de un vehículo en la ruta y se observará que cómo la credibilidad disminuye durante el trayecto mientras un vehículo k cumple con la demanda de n clientes.

Sea, la demanda de n clientes representada mediante el número difuso d_i y este a su vez se describe mediante los factores (d_{i1}, d_{i2}, d_{i3}) entonces,

³⁶ *Ibíd.*

$$CR_k = CI - \sum_{i=1}^n d_i$$

De este modo, CR_k (capacidad residual) es un número difuso de forma triangular:

$$CR_k = (CI - \sum_{i=1}^n d_{3i} \sum_{i=1}^n d_{2i} \sum_{i=1}^n d_{1i})$$

$$CR_k = (cr_{1k}, cr_{2k}, cr_{3k})$$

Con la capacidad residual del vehículo, fácilmente se obtiene el valor de la credibilidad Cre , la cual se traduce a su vez en el riesgo existente frente a que se pueda o no cumplir con la demanda del siguiente cliente:

$$Cre = cre\{CR_k \geq d_{n+1}\}$$

$$Cre = cre\{(d_{1,n+1} - cr_{1,n}, d_{2,n+1} - cr_{2,n}, d_{3,n+1} - cr_{3,n}) \leq 0\}$$

$$= \begin{cases} 1 & \text{si } d_{3,n+1} \leq cr_{1,n} \\ \frac{cr_{3,n} - d_{1,n+1}}{2(cr_{3,n} - d_{1,n+1} + d_{2,n+1} - cr_{2,n})} & \text{si } d_{1,n+1} \leq cr_{3,n}, d_{2,n+1} \geq cr_{2,n} \\ \frac{d_{3,n+1} - cr_{1,n} - 2(d_{2,n+1} - cr_{2,n})}{2(cr_{2,n} - d_{2,n+1} + d_{3,n+1} - cr_{1,n})} & \text{si } , d_{2,n+1} \leq cr_{2,n}, d_{3,n+1} \geq cr_{1,n} \\ 0 & \text{si } d_{1,n+1} \geq cr_{3,n} \end{cases}$$

Es así, como el valor de la credibilidad después de servir al cliente actual deberá de ser mayor o por lo menos igual a una ya dada por el experimentador para que el vehículo en ruta se dirija a servir al siguiente cliente. Sin embargo, si la

credibilidad es menor a este valor dado, el vehículo regresará al depósito dando así por terminado el recorrido.

Como en el caso del CVRP propuesto la restricción de capacidad del vehículo se maneja para cada ruta vehicular, del mismo modo la credibilidad se aplicará no para cada cliente en el transcurso del recorrido, sino para la totalidad de la ruta. Aplicando los conceptos de credibilidad en el modelo CVRP clásico se obtiene el siguiente modelo FVRP:

$$F.O.$$

$$Min z = \sum_i^n \sum_j^n \sum_k^p C_{ij} X_{ijk}$$

s. a.

$$Cre \left(\sum_i^n d_i Y_{ik} \leq CI \right) \geq Cre \star k = 1, 2, \dots, p.$$

$$\sum_i^n X_{ijk} = Y_{jk} \quad j = 0, 1, 2, \dots, n; k = 1, 2, \dots, p.$$

$$\sum_j^n X_{ijk} = Y_{ik} \quad i = 0, 1, 2, \dots, n; k = 1, 2, \dots, p.$$

$$\sum_i^n X_{ijk} = \sum_i^n X_{jik} \quad j = 0, 1, 2, \dots, n; k = 1, 2, \dots, p.$$

$$\sum_i^n \sum_k^p X_{ijk} = 1 \quad j = 0,1,2,\dots,n.$$

$$\sum_j^n X_{0jk} = 1 \quad k = 1,2,\dots,p.$$

$$\sum_i^n \sum_j^n C_{ij} X_{ijk} \leq L \quad k = 1,2,\dots,p.$$

$$X, Y \in [0,1], i, j = 0,1,2,\dots,n, k = 1,2,\dots,p.$$

En donde, $Cre \star \in [0,1]$ y simboliza el índice de preferencia del despachador, es decir, representa la actitud del distribuidor frente al riesgo de poder o no poder cumplir con la demanda del cliente siguiente. Dicho valor, como se ha mencionado con antelación, es dado por el experto, puesto que es necesario tener especial cuidado con éste parámetro. Si la relación $Cre \geq Cre \star$ se cumple, el vehículo k en ruta se dirigirá al siguiente cliente; de lo contrario el vehículo se dirigirá al depósito terminando con su recorrido y otro vehículo será quien cumpla con la siguiente demanda. De este modo, se puede decir que, el valor de $Cre \star$ influye directamente en las rutas de los vehículos y en la distancia total recorrida por cada uno durante su trayecto.

Otro factor de análisis que se evidencia al observar el modelo final, es qué valores bajos de $Cre \star$ incrementan el número de situaciones en las cuales un vehículo llega a un cliente y no puede cumplir con su demanda elevando así los costos logísticos de transporte, pues cabe la posibilidad que el vehículo hubiese podido atender otro cliente o simplemente antes de realizar este último recorrido haber regresado al origen.

Con el fin de comprender un mejor el manejo de la credibilidad (restricción de capacidad) y los conceptos vistos en esta sección se propone el siguiente ejemplo:

Se propone una red (ver Figura 11), en donde el nodo 1 será el punto origen y los demás poseen una demanda representada mediante números difusos triangulares de la forma $d_i = (d_{i1}, d_{i2}, d_{i3})$, entonces:

Si la capacidad inicial de cada vehículo k es CI , además $p = 3$ y $n = 7$ se tiene que:

$$Cre \left(\sum_i^n d_i Y_{ik} \leq CI \right) \geq Cre \star \quad k = 1,2,3$$

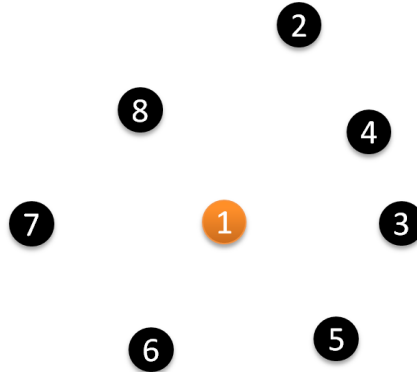
Aplicando el concepto de credibilidad visto, se podría escribir esta ecuación del siguiente modo:

$$Cre \left(\sum_i^n d_i Y_{ik} \leq CI \right) = \begin{cases} 0 & \text{si } CI \leq \sum d_{1i} Y_{ik} \\ \frac{CI - \sum d_{1i} Y_{ik}}{2(\sum d_{2i} Y_{ik} - \sum d_{1i} Y_{ik})} & \text{si } \sum d_{1i} Y_{ik} \leq CI \leq \sum d_{2i} Y_{ik} \\ \frac{-2 \sum d_{2i} Y_{ik} + \sum d_{3i} Y_{ik} + CI}{2(\sum d_{3i} Y_{ik} - \sum d_{2i} Y_{ik})} & \text{si } \sum d_{2i} Y_{ik} \leq CI \leq \sum d_{3i} Y_{ik} \\ 1 & \text{si } CI \geq \sum d_{3i} Y_{ik} \end{cases}$$

$$\geq Cre \star$$

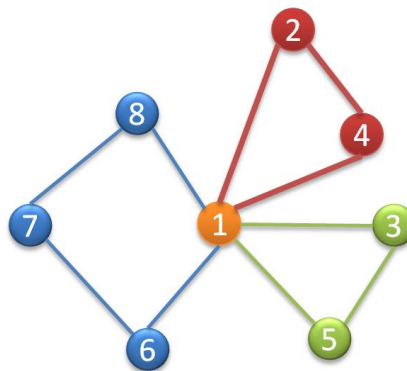
$$k = 1,2,3$$

Figura 11. Red de nodos



Ahora, si la credibilidad “*Cre*” generada de la suma de las demandas de un *subconjunto* de clientes (extraído del total de ellos) es mayor a la “*Cre*★” propuesta, quiere decir que a partir de dicho conjunto de clientes sería posible construir una ruta válida para el problema, por otro lado, si este criterio no se cumple se deben buscar aquellos que si lo logren. Si bien a partir de este criterio es posible generar subconjuntos de clientes válidos, el direccionamiento del vehículo (trayectoria) y otras condiciones del modelo estarán definidos por las demás restricciones.

Figura 12. Resultado de la red



En la Figura 12 se pueden apreciar tres agrupaciones de clientes, las cuales pueden traducirse en una posible respuesta para el ejemplo planteado. Ahora si se realiza un análisis sobre dicha respuesta, se afirma que se debe cumplir cada una de las siguientes condiciones:

$$(Cre(d_8 + d_7 + d_6) \leq CI) \geq Cre \star$$
$$k = 1$$

$$(Cre(d_2 + d_4) \leq CI) \geq Cre \star$$
$$k = 2$$

$$(Cre(d_3 + d_5) \leq CI) \geq Cre \star$$
$$k = 3$$

9. DESCRIPCIÓN DE HEURÍSTICAS Y METAHEURÍSTICAS PARA LA OPTIMIZACIÓN DEL MODELO FVRP

9.1. METAHEURÍSTICA DE OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS (PSO)

El PSO es un algoritmo metaheurístico enfocado a la optimización de resultados, es comúnmente utilizado en diversas líneas de la Investigación de operaciones y se encuentra fundamentado en el comportamiento de los enjambres de las abejas. El PSO fue propuesto en 1995 por los investigadores Kennedy y Eberhart³⁷, cuyo objetivo era el estudio de la colaboración en el desplazamiento de los organismos (aves, abejas, peces etc.), sin embargo durante los últimos años dicho algoritmo se ha logrado implantar exitosamente en la optimización de modelos de programación lineal y otras ramas de la ingeniería.

El objetivo de la optimización por enjambre de partículas (nube de partículas) es la determinación de la mejor posición que posea un organismo en una población, mediante el estudio de la ubicación y velocidad de aquellos que lo rodean, de este modo, análogamente en la investigación de operaciones se busca la mejor respuesta posible mediante el estudio de un conjunto de soluciones factibles denominadas partículas, las cuales poseen su propia posición y velocidad en un universo U , y por medio de ecuaciones matemáticas son trasladadas de un lugar a otro en el interior de este conjunto universo con el fin de ubicar aquella solución que cumpla en mayor medida las exigencias del experimentador.

³⁷ KENNEDY, James y EBERHART, Russell. Particle Swarm Optimization. En: Proceedings of the IEEE International Conference on Neural Networks. Noviembre-Diciembre, 1995. vol. 4, p. 1942–1948

Aunque la aplicación del PSO no garantiza que en todos los casos se obtenga la solución óptima del problema, una gran ventaja de este método metaheurístico es la poca carga computacional que posee, y cuya implantación en modelos matemáticos disminuiría del mismo modo la carga computacional que genera la resolución de estos modelos de manera exacta.

9.1.1. Ecuaciones del PSO

El PSO maneja dos ecuaciones, presentadas por Kennedy y Eberhart³⁸, una de posición de la partícula y otra de su velocidad:

- Ecuación de Posición:

$$X_t = X_{t-1} + V_t$$

- Ecuación de Velocidad:

$$V_t = (W * V_{t-1}) + ((Y_p * Ram) * (X_p - X_{ij})) + ((Y_g * Ram) * (X_g - X_{ij}))$$

La posición X_t simboliza la ubicación de la partícula en un tiempo t , V_t simboliza la velocidad de la partícula en un tiempo t , Y_p y Y_g son denominados grados de proximidad y según su valor determinaran la dirección de la velocidad de la partícula, X_p y X_g son la mejor posición local conocida y la mejor posición global conocida respectivamente, X_{ij} es la posición en cada partícula, Ram simboliza un número aleatorio que pertenece al conjunto $[0,1]$ y W es una constante de equilibrio.

³⁸ KENNEDY, James y EBERHART, Russell. Particle Swarm Optimization. Citado por: SALMAN, Ayed; AHMAD, Imtiaz y AL-MADANI, Sabah. Particle swarm optimization for task assignment problema. En: Microprocessors and Microsystems. Noviembre, 2002. vol. 26, no. 8, p. 363-371.

Para el uso de las ecuaciones del PSO es necesario en algunos casos trasladar los datos de un espacio discreto a uno continuo, a esto se le denomina codificar los datos, en estos casos se utiliza la ecuación que se enuncia a continuación, donde X_{max} y X_{min} son el rango de valores, Y_i es el número a modificar y n es la cantidad de números del conjunto.

- Ecuación de discreto a continuo:

$$X_{ij} = X_{min} + (X_{max} - X_{min}) * \frac{(Y_i - 1 + Ram)}{n}$$

Se tiene el conjunto discreto [5, 3, 1, 2, 4] y se desea obtener un conjunto continuo a partir de:

$$\text{Si } x_{max} = 5, x_{min} = 1 \text{ y } n = 5:$$

Conjunto continuo [4.15, 2.32, 0.31, 1.17, 2.8]

En un momento será necesario decodificar para obtener la ruta que genera el PSO, de este modo, para decodificar el conjunto lo único que hay que hacer es organizar los datos de menor a mayor, y colocar el dato de menor valor al número más pequeño en el conjunto original y el segundo menor al siguiente más pequeño, y así sucesivamente hasta que se decodifique completamente el número.

Se tiene el siguiente conjunto continuo a partir de la aplicación del PSO sobre un conjunto discreto:

Conjunto continuo [1.22, 3.1, 0.29, 2.37, 4.27]

Decodificado [2, 4, 1, 3, 5]

9.2. HEURÍSTICA DEL VECINO MÁS CERCANO

El vecino más cercano es una metodología básica de solución a los problemas de transporte dentro de la rama de programación lineal y optimización. Aunque las soluciones obtenidas comúnmente no son óptimas, esta heurística es una buena opción para la generación de conjuntos de soluciones factibles.

Para la aplicación de esta heurística, en un principio se parte de un punto origen y se observa cual es el nodo más próximo, de ahí se traza la ruta hasta él, luego, ubicado en el nodo se observa cual es el más cercano de los que faltan por visitar y se traza la ruta al siguiente nodo, este proceso se repite una y otra vez hasta haber visitado todos los nodos y por ultimo regresa al origen. Se puede ver con claridad que esta heurística en la mayoría de los casos no da una muy buena respuesta; sin embargo, para realizar el análisis sobre las rutas factibles y la aplicación de otras heurísticas y/o metaheurísticas para optimizar la respuesta, resulta ser una metodología bastante simple y eficaz.

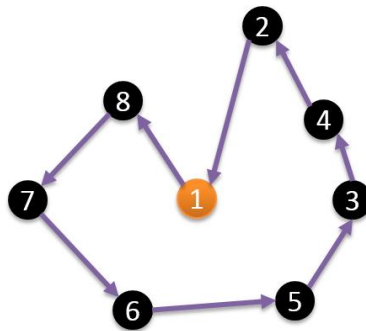
Un ejemplo sencillo se podría ver basado en la malla representada en la Figura 11, dado que se tiene la matriz de distancias establecidas del mismo (ver Tabla 3).

Tabla 3. Matriz de distancias

i\j	1	2	3	4	5	6	7	8
1	0	8	7	6	6	6	7	5
2	8	0	6	3	9	13	11	7
3	7	6	0	2	4	7	12	9
4	6	3	2	0	6	8	10	8

5	6	9	4	6	0	6	9	8
6	6	13	7	8	6	0	5	6
7	7	11	12	10	9	5	0	3
8	5	7	9	8	8	6	3	0

Figura 13. Solución por medio del vecino más cercano



En el resultado del ejemplo (ver

Figura 13) se aprecia con claridad la aplicación de los pasos vistos anteriormente. En caso que la capacidad del vehículo sea insuficiente para atender a todos los clientes, simplemente se toma este resultado y se colocan los reabastecimientos en los momentos en los cuales se tienen que dar.

10. CONSTRUCCIÓN DEL FRAMEWORK EN MATLAB DEL MODELO FVRP

10.1. DESCRIPCIÓN Y ANÁLISIS DEL ALGORITMO PROPUESTO EN MATLAB

El framework está basado en el modelo descrito en el capítulo 8 de este documento, sin embargo, encontrar un resultado completamente exacto del mismo requiere una carga computacional muy grande, lo cual genera tiempos de respuesta demasiado elevados. Esto se debe primeramente a la cantidad de datos procesados por el modelo y también debido a que Matlab® no es un software que se especializa en la resolución de estos problemas, de este modo, se hace necesario aplicar heurísticas y metaheurísticas además de estrategias de programación enfocadas a reducir el tiempo de respuesta y a su vez logren soluciones de calidad.

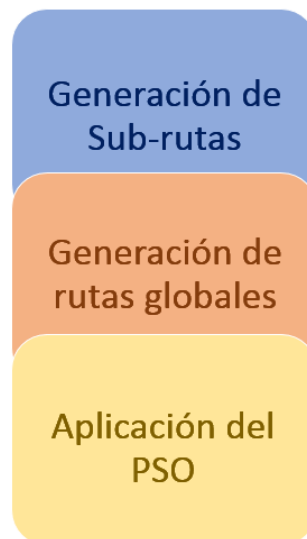
La programación del modelo está dividida en tres sectores como se ve en la Figura 14, los parámetros iniciales son extraídos desde una hoja de cálculo de MS Excel ubicada en el mismo directorio en el cual se encuentra el archivo de Matlab® con el nombre de "*Parámetros*". Del mismo modo los resultados son almacenados en un archivo también de MS Excel, el cual es creado por el mismo modelo y tiene el nombre de "*Resultados*". Es necesario que el archivo MS Excel de *Parámetros* exista para que el modelo pueda ejecutarse correctamente de lo contrario generará un error desde el inicio.

En dado caso, si el modelo es ejecutado por más de una vez, la hoja MS Excel de *Resultados* será reemplazada por una nueva, la cual contiene los datos respuesta del modelo corrido por última vez.

10.1.1. Generación de sub-rutas

Debido a que el PSO necesita de la generación de un conjunto de resultados factibles (partículas), se utiliza la restricción de capacidad, que para el caso también es la restricción de credibilidad, con el fin de generar conjuntos de clientes que el vehículo K se crea pueda visitar y cumplir con su demanda en un solo recorrido sin que se exceda la capacidad del mismo durante su trayecto. Posteriormente se aplica la heurística del Vecino más cercano para construir a partir de los conjuntos de clientes sub-rutas viables para el proceso es decir, los conjuntos de clientes pasan a ser conjuntos de rutas para un solo vehículo. En este punto, se aplica a estas rutas la restricción de distancia máxima, con el fin de eliminar aquellas rutas que tengan un desplazamiento muy largo y que un vehículo no logre hacer en un solo recorrido.

Figura 14. Sectores del algoritmo programado



Ahora, se decide implantar un parámetro el cual se denomina "*Mínimo número de clientes a atender*", el objetivo de la introducción de este parámetro es que el

experimentador pueda decidir cuál va a ser el número mínimo de nodos que un solo vehículo pueda visitar, de este modo eliminar algunas rutas creadas por los pasos anteriores y que no son necesarias para el modelo.

Un inconveniente de este sector del modelo es la incapacidad de la memoria del computador para almacenar grandes cantidades de rutas; debido a éste inconveniente se debe tener cuidado al ingresar al modelo los parámetros generales, dado que si se generan gran cantidad de rutas individuales es posible que el modelo genere un error de memoria.

10.1.2. *Generación de rutas globales (Partículas)*

Una vez completados los pasos anteriores, se tiene un conjunto de rutas las cuales pueden darse para un solo vehículo, con el cual se dispone a construir el conjunto de rutas globales para la aplicación del PSO. Las rutas obtenidas son de este tipo [2, 5, 3], significa que algún vehículo podrá ir de 1 a 2, de 2 a 5 y de 5 a 3 para luego regresar al origen (1), esta sería la ruta de tan solo un vehículo. En el caso que se tuvieran dos vehículos, para construir la ruta global, es necesario buscar en el conjunto de rutas aquellas o aquella que se complementen y logren cumplir con la demanda de los n clientes, es decir, si se toma la ruta [2,5,3] y son sólo 5 clientes (2, 3, 4, 5, 6), entonces deberá haber unas rutas que contengan los clientes restantes, un ejemplo sería [4, 6], por consiguiente una ruta global podría verse desde esta forma [2, 5, 3 | 4, 6], de este modo se generan las rutas globales requeridas por el experimentador para posteriormente aplicar el PSO.

Un inconveniente en este paso es el tiempo de ejecución, debido a que se deben realizar una gran cantidad de iteraciones en la búsqueda de las rutas que cumplan con los requisitos y poder almacenar las mejores respuestas. Para reducirlo se implementan estrategias que eviten iteraciones innecesarias; a pesar de esto, el

tiempo no mejora de manera considerable, así que se opta por implementar un parámetro de tiempo. Dicho parámetro establece un tiempo máximo de generación de rutas globales (*Tiempo máximo*), de forma que cuando se utilice el modelo, se pueda detener este proceso en algún punto y utilizar las mejores rutas que se hayan logrado obtener hasta el momento.

Se podría decir que este parámetro de tiempo decide en gran medida la calidad de las rutas globales que se utilizarán posteriormente en el PSO, esto debido a que la programación lo que busca es encontrar las mejores n rutas: entre más tiempo el modelo tenga para generar las rutas globales es muy probable que la calidad de éstas sea mucho mejor.

10.1.3. *Aplicación del PSO*

El tercer sector del modelo, aplica la heurística PSO al conjunto de rutas obtenido por medio de los dos pasos anteriores. Una consideración que se tiene en este sector, es mantener la posición de las partículas durante las interacciones, aunque estas posiciones no se traducen en rutas factibles para el modelo. Sin embargo, las rutas que se mantengan guardadas serán las mejores posiciones locales entre las cuales deberá estar la mejor global, y solo serán alteradas si en alguna iteración surge una posición para la partícula factible que sea mejor.

Cada uno de los sectores del modelo pueden ser identificados fácilmente dentro del mismo, ya que se hizo un escrito referente a su función antes de cada uno, además de estos sectores, se pueden también observar otras funciones que hacen diferentes tipos de programaciones. En la Imagen 1 y la Imagen 2 se mostraran algunos sectores de la programación del modelo en el software Matlab®.

Imagen 1. Sectores del modelo programado

```
164 %% Vecino mas cercano
165 %% se generan las rutas correspondientes a cada Y que cumpla
166 %% la condicion anterior utilizando la metadaturistica del vecino mas cercano
167 if vr7<1
168     c_yc=0;
169     for i=1:i_1
170         if Y(i,1)>0
171             c_yc=c_yc+1;
172             Yc(c_yc,1)=i;
173         end
174     end
175
176     Ru=zeros(1,i_1+1);
177     Ru(1)=1;
178     tmp=0;
179
180     for q=1:c_yc-1
181         qq=q-1;
182         condicion=L;
183         for d=2:c_yc-qq
184             if C(Ru(q),Yc(d))<condicion
185                 condicion=C(Ru(q),Yc(d));
186                 tmp=d;
187             end
188         end
189     end
```

Imagen 2. Sectores del modelo programado

```
FVRP_v4.m
271     clientes(k)=ca_x((k_1-(k-1)));
272 end
273
274 while num2(k_1+1)<1
275     k=1;
276     num2
277     %%se generan todas las posibles combinaciones de las rutas existentes
278     %%para los k vehiculos a modo de obtener el conjunto total de rutas
279     %%necesarias para atender los clientes
280     while num2(k)>(nrutas-k)
281         num2(k+1)=num2(k)+1;
282
283         clientes(k+1)=ca_x(num2(k+1));
284         Y(1:i_1,1,k+1)=Yr(1:i_1,1,num2(k+1));
285         X(1:i_1,1:j_1,k+1)=Xr(1:i_1,1:j_1,num2(k+1));
286         Xi(k+1,1:i_1)=Xl(1,1:i_1,num2(k+1));
287
288         for ik=k:-1:1
289             num2(ik)=num2(ik)+1;
290             Y(1:i_1,1,ik)=Yr(1:i_1,1,num2(ik));
291             X(1:i_1,1:j_1,ik)=Xr(1:i_1,1:j_1,num2(ik));
292             clientes(ik)=ca_x(num2(ik));
293             Xi(ik,1:i_1)=Xl(1,1:i_1,num2(ik));
294         end
295         k=k+1;
296     end
297
298     Y(1:i_1,1,1)=Yr(1:i_1,1,num2(1));
299     X(1:i_1,1:j_1,1)=Xr(1:i_1,1:j_1,num2(1));
300     clientes(1)=ca_x(num2(1));
301     Xi(1,1:i_1)=Xl(1,1:i_1,num2(1));
302     num2(1)=num2(1)+1;
303
304     cont12=cont12+1;
305
306     cls=sum(clientes);
307     if (cls==c1)
308         %% restricción4
309         %%se verifica que las rutas seleccionadas atiendan todos los clientes sin
```

MATRO Laboratory

11. MANUAL DE USO DEL FRAMEWORK

Los parámetros generales del modelo se ingresan al mismo por medio de un archivo de MS Excel, ubicado en el mismo directorio que el archivo de Matlab®, los parámetros se pueden modificar fácilmente y se dividen en las tres hojas del mismo archivo de MS Excel.

En la primera hoja de cálculo se pueden modificar todos los parámetros que involucran las dimensiones del modelo a tratar, en primer lugar se encuentra la cantidad de nodos que va a tener el modelo que se identifican con las letras “ j_l ”, y “ i_l ”, refiriéndose a la longitud de “ j ” y de “ i ” respectivamente. El parámetro “ k_l ” indica el número de vehículos que tendrá el modelo. “ L ” simboliza la longitud máxima a recorrer de cada vehículo y “ C_l ” la capacidad inicial de los mismos. “*Credibilidad*”, como su nombre lo indica, es la credibilidad con la cual se van a construir las rutas. El parámetro “ n_l ” es el mejor número n de rutas encontradas por el algoritmo antes de aplicar el PSO y las cuales se guardan mediante el paso 2 del mismo modelo. “ l ” es el número de iteraciones que realizará el PSO antes de almacenar los resultados. También se encuentran “ Y_p ”, “ Y_g ” y “ w ”: los parámetros de las fórmulas utilizadas en el PSO.

Adicionalmente, “*rutas a descartar*” es un parámetro que elimina del número de rutas iniciales, todas aquellas que para el concepto del experimentador, son rutas que cumplen con un número de clientes lo suficientemente pequeño para no ser tomadas en cuenta dentro del desarrollo del modelo, es decir, si se tienen dos vehículos, 10 clientes y las demandas de estos últimos son muy parecidas entre ellas, una ruta factible para uno de ellos sería atender en su recorrido a 2 clientes ya que no excedería la capacidad del mismo y la distancia recorrida es menor a la máxima, sin embargo esto implicaría que el otro vehículo no tendría la capacidad

suficiente para atender con los 8 clientes restantes y ninguna ruta asignada a este cumpliría con la totalidad de la demanda.

Por último, “*tiempo máximo*” es un parámetro que se agregó al sector de combinación de rutas, con el fin de dar un límite de tiempo a este proceso, dado que los tiempos de combinación en ciertos casos son demasiado elevados si se realizaran todas las iteraciones posibles. Debido a esto y a que el PSO no requiere de una gran cantidad de rutas sino solo de algunas de ellas, se decidió incluir dicho parámetro. En la Imagen 3 se observa la captura de pantalla con los parámetros generales del modelo en su respectiva hoja de cálculo.

Imagen 3. Parámetros generales del modelo

	A	B	C	D	E	F
1	j_l	18				
2	i_l	18				
3	k_l	3				
4	L	275				
5	Cl	160				
6	Credibilidad	0,3				
7	nr	80	Numero de rutas a guardar			
8	t	750	Numero de interacciones PSO			
9	Yp	0,2				
10	Yg	1,5				
11	w	0,1				
12	rutas a descartar	4				
13	tiempo maximo [m]	600				
14						
15						
16						

En las hojas 2 “*Distancias*” y 3 “*Demandas*” (ver Imagen 4 e Imagen 5) del documento de MS Excel “*parámetros*”, se encuentran la matriz de distancias y las demandas difusas triangulares respectivamente.

Es necesario aclarar, que los valores de ir de un lugar a ese mismo lugar se toman como valores elevados para el problema, con el fin de que al correr el algoritmo

este no los tome como rutas factibles, es por esto que en la Imagen 4 se observa que el valor o la distancia de ir de un punto al mismo es 999 y dependiendo del problema será necesario elevar este valor.

Imagen 4. Matriz de distancias entre nodos

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
1	999	45	41	30	28	68	21	64	26	48	44	35	41	57	26	48	25	66	43	60	49	2	
2	45	999	86	52	22	95	67	94	19	65	67	53	67	91	66	64	25	110	47	105	94	1	
3	41	86	999	56	67	58	20	52	67	58	60	54	47	39	28	69	65	26	68	19	11	7	
4	30	52	56	999	46	96	39	92	38	77	16	64	71	83	54	18	27	82	70	72	67	3	
5	28	22	67	46	999	74	48	73	9	46	62	33	46	70	44	62	21	90	30	86	73	1	
6	68	95	58	96	74	999	62	6	81	31	107	42	28	19	43	114	88	56	51	65	51	8	
7	21	67	20	39	48	62	999	57	48	52	47	43	42	46	20	55	45	46	55	39	29	5	
8	64	94	52	92	73	6	57	999	79	31	103	41	27	13	39	110	85	50	52	59	45	8	
9	26	19	67	38	9	81	48	79	999	54	54	40	53	75	48	54	12	91	39	86	75		
10	48	65	58	77	46	31	52	31	54	999	91	14	11	35	32	96	62	69	21	73	57	€	
11	44	67	60	16	62	107	47	103	54	91	999	79	84	93	65	10	43	85	86	73	72	5	
12	35	53	54	64	33	42	43	41	40	14	79	999	15	41	26	83	48	70	14	71	56	4	
13	41	67	47	71	46	28	42	27	53	11	84	15	999	26	22	89	59	59	27	62	46	5	
14	57	91	39	83	70	19	46	13	75	35	93	41	26	999	30	100	79	38	54	46	32	€	
15	26	66	28	54	44	43	20	39	48	32	65	26	22	30	999	71	50	46	40	45	30	5	
16	48	64	69	18	62	114	55	110	54	96	10	83	89	100	71	999	42	95	88	83	81	4	
17	25	25	65	27	21	88	45	85	12	62	43	48	59	79	50	42	999	90	50	84	74		
18	66	110	26	82	90	56	46	50	91	69	85	70	59	38	46	95	90	999	84	14	17	9	
19	43	47	68	70	30	51	55	52	39	21	86	14	27	54	40	88	50	84	999	85	70	4	
20	60	105	19	72	86	65	39	59	86	73	73	71	62	46	45	83	84	14	85	999	16	8	
21	49	94	11	67	73	51	29	45	75	57	72	56	46	32	30	81	74	17	70	16	999	7	
22	29	18	70	34	16	88	50	86	7	61	50	47	59	81	53	49	7	95	46	89	78	99	
23	53	56	59	71	46	73	53	61	88	67	34	39	34	73	98	75	54	60	76	42	60		
24	45	56	56	63	41	70	48	53	87	59	28	32	38	72	91	69	46	52	69	34	53	1	
25	74	54	36	56	42	26	35	49	55	70	90	52	84	50	41	22	71	56	29	79	35	€	
26	22	97	88	46	64	97	70	48	125	32	23	52	89	110	81	75	25	41	70	21	61	6	
27	52	50	31	37	24	30	18	28	63	50	68	32	73	53	36	8	50	35	14	57	14	7	
28	81	62	72	98	67	86	73	87	91	95	60	64	27	78	123	98	82	87	100	70	83	2	
29	87	21	39	92	50	50	52	79	46	96	79	56	19	34	105	76	86	83	81	80	68	4	
30	60	72	52	37	45	47	40	35	79	52	82	50	95	72	16	15	57	41	14	68	30	9	
31																							
32																							
33																							
34																							

Como se ha mencionado anteriormente, los resultados que son arrojados al correr el modelo son almacenados en un archivo con el nombre “*resultados*”. En las primeras dos hojas del documento están contenidos los resultados finales del modelo: en la primera hoja se almacenan las rutas de cada vehículo y en la segunda el valor de la función objetivo asociado a los mismos respectivamente. En las hojas tres y cuatro se encuentran las rutas y los resultados de aquellas a las que se les aplico el PSO es decir, son las n rutas que se generaron por los pasos 1 y 2 para usarse en dicha metaheurística. En la Imagen 6 se puede ver las matrices de resultados de un ejercicio terminado. Cabe resaltar que el número de

vehículos de éste ejemplo es 3, razón por la cual se agruparon las rutas en grupos de tres tanto para la hoja 1 como para la hoja 3.

Imagen 5. Matriz de demandas Difusas triangulares

	A	B	C	D
1	0	0	0	
2	4	6	8	
3	18	21	23	
4	25	28	31	
5	29	32	35	
6	14	16	18	
7	20	22	24	
8	31	32	33	
9	4	7	9	
10	17	18	19	
11	31	34	37	
12	12	13	14	
13	35	37	39	
14	14	17	19	
15	10	12	14	
16	3	6	9	
17	11	13	14	

Imagen 6. Rutas después del PSO

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
4	1	11	16	14	18	6	8	12	1	0	0	0	0	0	0	0	0	0
5	1	9	10	15	13	17	1	0	0	0	0	0	0	0	0	0	0	0
6	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0
7	1	11	16	14	18	6	8	12	1	0	0	0	0	0	0	0	0	0
8	1	9	10	15	13	17	1	0	0	0	0	0	0	0	0	0	0	0
9	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0
10	1	11	16	14	18	6	8	12	1	0	0	0	0	0	0	0	0	0
11	1	9	10	17	13	15	1	0	0	0	0	0	0	0	0	0	0	0
12	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0
13	1	11	16	14	18	6	8	1	0	0	0	0	0	0	0	0	0	0
14	1	12	9	10	15	13	17	1	0	0	0	0	0	0	0	0	0	0
15	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0
16	1	11	16	14	18	6	8	1	0	0	0	0	0	0	0	0	0	0
17	1	12	9	10	15	13	17	1	0	0	0	0	0	0	0	0	0	0
18	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0
19	1	11	16	14	18	6	8	1	0	0	0	0	0	0	0	0	0	0
20	1	12	9	10	15	13	17	1	0	0	0	0	0	0	0	0	0	0
21	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0

Los resultados son almacenados de menor a mayor. En la Imagen 7 se observa que hay varios resultados con el mismo valor, esto se da debido a que varias rutas

lograron convertirse en la mejor global o simplemente se acercaron a esta convirtiéndose en una mejor.

Imagen 7. Resultados después del PSO

	A	
1	659,00937	
2	659,00937	
3	659,00937	
4	669,57346	
5	671,33751	
6	671,33751	
7	671,33751	
8	671,33751	
9	671,33751	
10	674,07408	
11	678,687	
12	684,42688	
13	684,72665	
14	689,56173	
15	692,59697	
16	694,07711	

En la Imagen 8 y la Imagen 9 se pueden ver tanto las rutas como los resultados de las partículas usadas en el PSO, cada uno ellos es diferente dado que restaría efectividad al PSO que existiesen rutas iguales. Por otro lado el PSO, en este caso, logra encontrar dos rutas mejores para dicho experimento y acerca otras a estas dos mejores. Es importante mencionar que no en todos los casos el PSO logra encontrar una mejor respuesta, esto se da porque:

- Lo buenas que sean las rutas dadas al PSO (es decir las “*partículas*”), debido a que sería más difícil para éste encontrar una mejor respuesta si ya se ha encontrado una muy buena.
- Las rutas mejores no podían ser interceptadas por la trayectoria de las partículas dadas.

Imagen 8. Rutas antes del PSO

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
4	1	12	18	10	15	13	17	1	0	0	0	0	0	0	0	0	0	0
5	1	9	11	16	14	6	8	1	0	0	0	0	0	0	0	0	0	0
6	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0
7	1	9	18	10	15	13	17	1	0	0	0	0	0	0	0	0	0	0
8	1	12	11	16	14	6	8	1	0	0	0	0	0	0	0	0	0	0
9	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0
10	1	12	9	14	18	10	6	8	1	0	0	0	0	0	0	0	0	0
11	1	11	16	17	15	13	1	0	0	0	0	0	0	0	0	0	0	0
12	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0
13	1	12	8	18	10	15	13	17	1	0	0	0	0	0	0	0	0	0
14	1	9	11	16	14	6	1	0	0	0	0	0	0	0	0	0	0	0
15	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0
16	1	11	16	14	18	6	8	1	0	0	0	0	0	0	0	0	0	0
17	1	12	9	10	15	13	17	1	0	0	0	0	0	0	0	0	0	0
18	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0
19	1	12	18	10	15	13	17	1	0	0	0	0	0	0	0	0	0	0
20	1	9	5	7	11	16	14	1	0	0	0	0	0	0	0	0	0	0
21	1	4	2	8	6	3	1	0	0	0	0	0	0	0	0	0	0	0

Imagen 9. Resultados antes del PSO

H10	
	A
1	671,33751
2	674,07408
3	678,687
4	684,72665
5	689,56173
6	692,30566
7	692,59697
8	694,07711
9	694,24895
10	700,64433
11	702,30499
12	703,81503
13	704,18659
14	704,59631
15	704,92258
16	705,27827

12. DISEÑO DE EXPERIMENTOS DEL MODELO PROGRAMADO

Para verificar el correcto funcionamiento del modelo y obtener un mejor manejo de él, se decide realizar un estudio del modelo mediante un diseño de experimentos; de este modo es necesario obtener conocimiento de qué parámetros impactan de mayor forma la variable respuesta y a su vez determinar qué valor usar para optimizar dicho resultado. El diseño de experimentos a realizar al modelo es un diseño 2^{k-p} , en el cual se realizaron 8 corridas experimentales. Los computadores utilizados en este proceso son Intel Core i7, además la versión de Matlab® usada es la 2014^a.

En la Tabla 4, se observa los factores que se tiene en cuenta para el diseño de experimentos a realizar, además se pueden ver cómo fueron manejados durante el proceso. Por otra parte la Tabla 5 muestra los datos de entrada del experimento, es decir, los resultados de cada una de las 8 corridas realizadas.

Tabla 4. Factores del diseño experimental

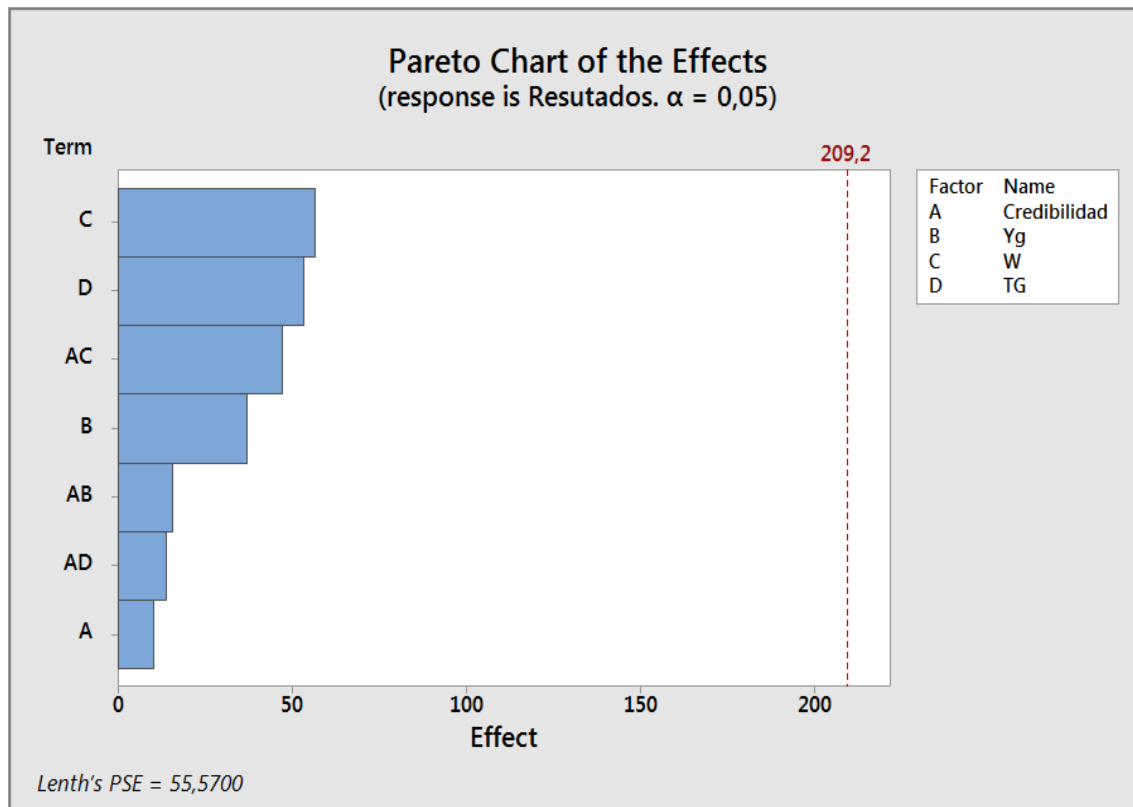
	+	-
Credibilidad	0,9	0,3
W	0,9	0,1
Yg	1,5	0,3
Tg (<i>Tiempo máximo</i>)	1440	60

En base a los resultados obtenidos, mediante el software Minitab 17, se generan las siguientes gráficas, en las que se pueden ver los impactos de los parámetros *TG*, *w*, *Cre* y *Yg*, además de las interacciones entre estos. Mediante estas graficas es posible determinar cómo se deben manejar cada uno de los factores para así optimizar la respuesta del modelo.

Tabla 5. Datos de entrada para el diseño experimental

564,080868
629,320951
659,009372
734,918619
615,477198
578,613504
646,851834
598,200641

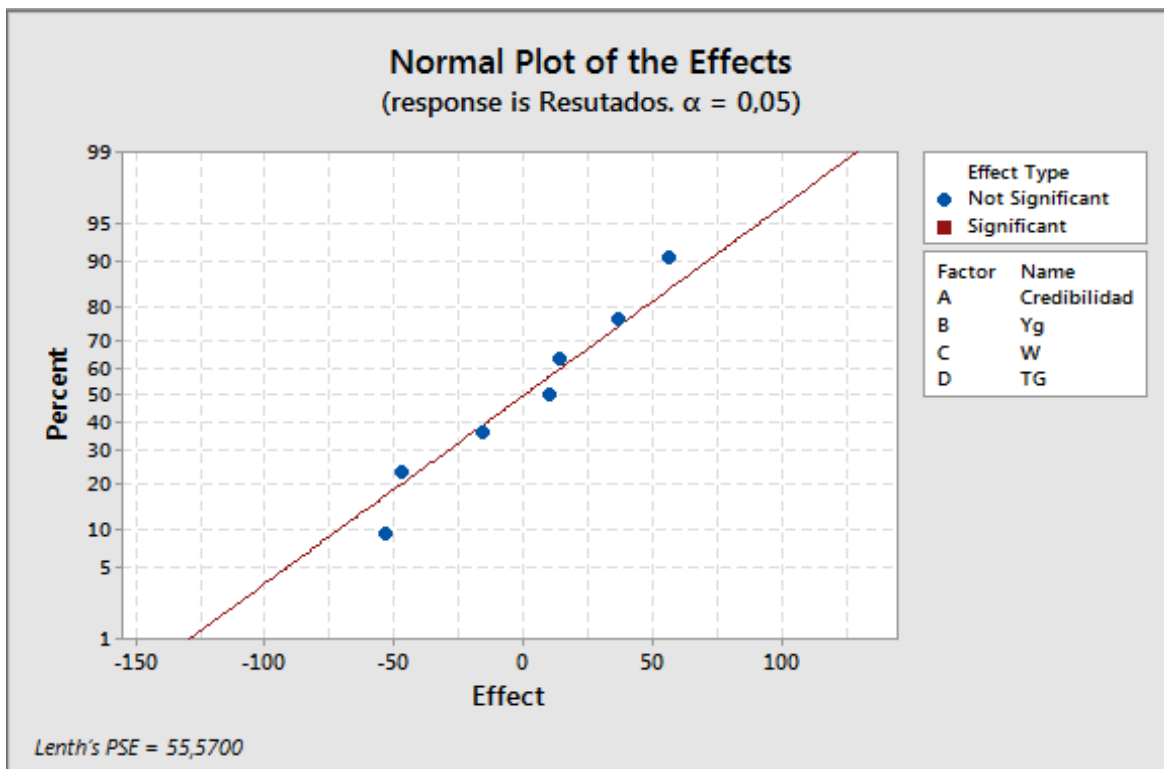
Figura 15. Diagrama de Pareto para los efectos



Mediante la Figura 15 y la Figura 16 se puede observar el impacto de cada uno de los factores sobre la variable respuesta. Los factores “TG” y “w” junto con la

interacción de la “credibilidad” y “w” fueron los que tuvieron un mayor impacto sobre la respuesta, sin embargo el factor credibilidad, que es el que posee el parámetro difuso no tuvo por si solo mayor influencia en el experimento.

Figura 16. Gráfica de normalidad del experimento

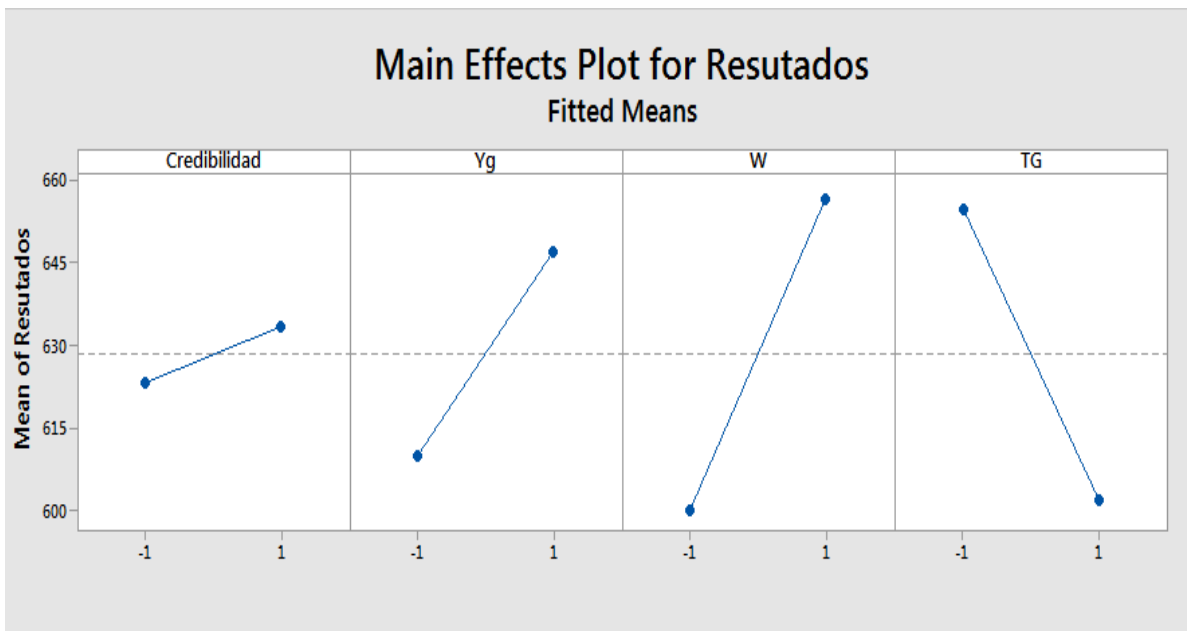


El valor de la credibilidad aunque no tiene impacto que los demás factores, si se realiza un análisis más profundo, sí debe tener un impacto considerable; esto se debe a que al observar el número de rutas generadas con un valor alto de credibilidad y las generadas con un valor bajo la diferencia no es mucha, en parte debido a que estos resultados son optimizados por el PSO. No obstante, se puede deducir con facilidad que al tener valores bajos de credibilidad los vehículos podrían llegar a visitar más clientes por ruta que con valores altos de dicho parámetro.

Continuando con el análisis, la Figura 16 (Gráfica de normalidad) corrobora al mismo tiempo la veracidad del experimento y a su vez ayuda a determinar en ciertos casos el impacto de los efectos sobre la respuesta.

Ahora, la Figura 17 y la Figura 18 ayudan a determinar la manera óptima de manejar los factores: por ejemplo el valor de “w” funciona mejor o impacta mejor la respuesta cuando es menor, dado que las rutas generadas por los modelos cuando “w” estaba en bajo eran más óptimas. Claramente se puede ver que con el parámetro “TG” sucede lo contrario, cuando este valor es mayor las respuestas arrojadas por el modelo tienden a ser mucho mejores.

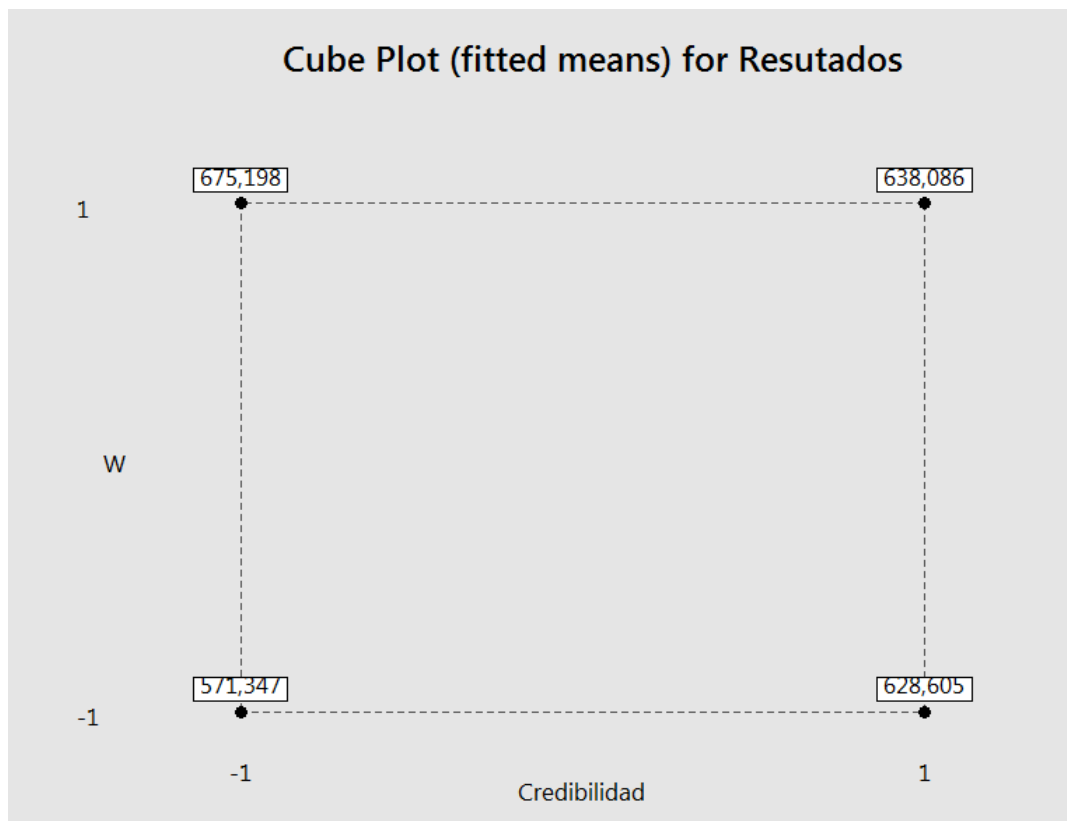
Figura 17. Grafica de efectos para cada factor



La grafica de cubos (ver Figura 18) sirve para determinar que los parámetros “w” y “Credibilidad” deben de manejarse bajos, si se desean obtener rutas que se traduzcan en bajos costos. Aun así, es necesario tener un cuidado especial con el manejo de la credibilidad, dado que como se ha dicho con anterioridad, valores

bajos de este parámetro incrementarían el número de veces que un vehículo llegue a un cliente y no tenga capacidad suficiente para atender la demanda del mismo. En resumen, se podría decir que los parámetros a manejar con valores bajos serían w , Yg y *credibilidad*, mientras que Tg sería el factor a manejarse en alto, esto debería de ser así si se quiere obtener resultados con bajos costos.

Figura 18. Grafica de cubos de la interacción de los efectos *Cre* y w



Para verificar los resultados obtenidos por los análisis hechos anteriormente se realizó un experimento nuevamente manejando los parámetros según lo estipulado, y evidentemente al contrastar el resultado encontrado con los del diseño de experimentos, este era menor (ver Tabla 6).

Tabla 6. Contraste de resultados

Resultados antes del diseño de experimentos	Resultado después del diseño de experimentos
564,080868	541,8136093
629,320951	
659,009372	
734,918619	
615,477198	
578,613504	
646,851834	
598,200641	

13. VERIFICACIÓN DEL MODELO MEDIANTE INSTANCIAS

Como se ha podido ver, el modelo programado ofrece una nueva alternativa de solución a sistemas de ruteo. Sin embargo, este debe responder correctamente a diferentes tipos de problemas que se puedan llegar a dar durante su uso práctico en el mercado. De este modo, se debe verificar su comportamiento frente a diferentes tipos de situaciones (instancias), para observar los resultados arrojados por dicho modelo frente a cada una de estas y verificar su funcionalidad.

Como en la vida real las demandas son valores reales y no difusos, es necesario dar alternativas para transformar dicho valor en un número difuso triangular. Como se explicó a través del documento, el modelo se propuso con el fin de dar soluciones factibles a situaciones donde la demanda no se conoce con certeza, es así, como es necesario para la aplicación del programa construido tener un supuesto o historial del cual se pueda extraer dicha función difusa que simbolice la demanda.

Para el análisis que se verá a continuación, se propusieron unos historiales de demanda, y a partir de ellos se construyeron números difusos triangulares (a_1, a_2, a_3) (ver tablas 9, 12, 15, 18, 21, 24, 27, en el que a_1 fue el menor valor dentro del historial y a_3 el mayor, mientras que a_2 simboliza el valor de demanda con mayor grado de ocurrencia dentro del historial (ver Figura 19).

En la Imagen 10, se ve cómo se tomaron los parámetros generales para cada uno de los escenarios planteados. Los parámetros faltantes son diferentes para cada escenario dado. Por otra parte, se decidió tomar el parámetro de credibilidad de 0.8, esto con el fin de disminuir el riesgo, de tal modo se minimicen el número de situaciones en las que en realidad el vehículo llegue al cliente y no logre cumplir con su demanda.

Figura 19. Demanda difusa triangular

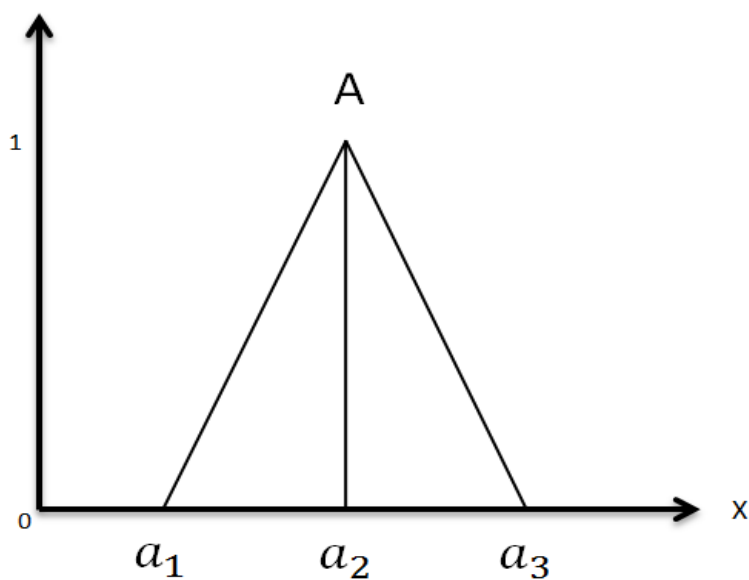


Imagen 10. Parámetros generales de los escenarios

j_l				
i_l				
k_l				
L				
CI				
Credibilidad	0,8			
nr	80		Numero de rutas a guardar	
t	500		Numero de interacciones PSO	
Y_p	0,2			
Y_g	0,3			
w	0,1			
rutas a descartar				
tiempo maximo [m]	1440			

A continuación se verán cada una de las instancias o escenarios planteados con sus resultados respectivos arrojados por el modelo propuesto.

13.1. ESCENARIO 1 (8 CLIENTES & 1 VEHÍCULO)

Con este escenario se busca determinar la calidad de la ruta generada para un solo vehículo, sin embargo como se observa el resultado no es en muchos casos el mejor dado que solo se genera una ruta global antes del PSO y por tanto esta será la mejor y única ruta generada por el modelo.

La capacidad del vehículo y su distancia máxima a recorrer son 230 y 300 respectivamente. Los resultados de esta instancia se pueden ver en la Tabla 9.

Tabla 7. Posiciones de los nodos escenario 1

x	y
33	55
33	60
14	27
88	57
55	51
12	87
69	5
94	39

Tabla 8. Demandas del Escenario 1

Demandas		
1	4	15
20	20	34
17	19	29
6	15	19
18	28	35
5	15	17

94

4	13	14
12	15	29

Tabla 9. Resultado Escenario 1

Vehículos	Rutas										Distancia
1	1	2	5	4	8	7	3	9	6	1	283,12

13.2. ESCENARIO 2 (8 CLIENTES & 2 VEHÍCULOS)

La capacidad de los vehículos y su distancia máxima a recorrer son 250 y 450 respectivamente. Los resultados de esta instancia se pueden ver en la Tabla 12.

Tabla 10. Posiciones de los nodos escenario 2

x	y
144	163
98	83
52	136
126	80
60	194
196	50
68	80
155	149
68	66

Tabla 11. Demanda escenario 2

Demandas		
15	21	51
19	37	64
21	32	43
17	36	45
25	47	72
28	44	57
27	43	62

37	62	83
----	----	----

Tabla 12. Resultado Escenario 2

vehículos	Rutas							Distancia
1	1	7	2	9	3	5	1	690,26
2	1	6	4	8	1			

De este escenario en adelante, ya se generan una cantidad considerable de rutas globales para que el PSO pueda trabajar en su optimización.

13.3. ESCENARIO 3 (14 CLIENTES & 2 VEHÍCULOS)

La capacidad de los vehículos y su distancia máxima a recorrer son 280 y 1900 respectivamente. Los resultados de esta instancia se pueden ver en la Tabla 15.

Tabla 13. Posiciones de los nodos escenario 3

x	y
398	73
405	102
181	85
274	174
316	491
355	486
151	399
421	241
63	178
411	205
97	470
326	276
498	372
172	406
109	488

Tabla 14. Demanda escenario 3

Demanda		
17	19	23
8	11	17
13	18	26
25	31	37
27	37	47
15	25	25
10	16	26
10	13	13
17	21	21
16	26	34
13	17	26
22	29	37
18	27	36
18	27	35

Tabla 15. Resultado Escenario 3

vehículos	Rutas										Distancia
1	1	10	8	12	13	14	11	15	9	1	2847,618576
2	1	2	4	3	7	5	6	1			

13.4. ESCENARIO 4 (14 CLIENTES & 3 VEHÍCULOS)

La capacidad de los vehículos para este escenario es de 110 y la distancia máxima de recorrido de cada uno es de 610. El número mínimo de clientes a atender por cada vehículo es de 2.

Tabla 16. Posiciones de los nodos escenario 4

x	y
99	222
134	315

255	174
196	114
369	136
224	181
167	297
149	228
348	209
357	160
257	246
150	123
182	311
306	150
233	156

Tabla 17. Demanda escenario 4

Demandas		
9	10	17
6	14	17
7	19	26
9	17	20
5	17	21
6	15	18
7	12	15
10	16	22
8	19	20
10	18	25
5	7	8
7	13	17
5	16	18
9	15	20

Tabla 18. Resultado Escenario 4

Vehículos	Rutas								Distancia
1	1	8	6	15	5	10	3	1	1479,527969
2	1	12	4	14	9	11	1		
3	1	2	7	13	1				

13.5. ESCENARIO 5 (17 CLIENTES & 3 VEHÍCULOS)

La capacidad de los vehículos para este escenario es de 340 y la distancia máxima de recorrido de cada uno es de 2250. El número mínimo de clientes a atender por cada vehículo es de 3.

Tabla 19. Distancias entre nodos escenario 5

x	y
335	504
727	268
618	714
275	258
744	534
334	377
790	695
626	335
228	147
228	747
699	223
796	288
484	388
705	541
228	228
624	383
377	179
94	504

Tabla 20. Demanda escenario 5

Demanda		
20	30	32
47	66	94
30	48	50
24	41	43
17	35	74

15	20	62
40	64	67
50	70	96
6	10	23
12	30	48
49	50	73
20	30	49
38	41	51
35	53	81
12	42	59
48	60	69
9	39	47

Tabla 21. Resultados escenario 5

Vehículos	Rutas								Distancia
1	1	13	16	2	11	12	14	1	4105,321416
2	1	10	18	4	15	9	17	1	
3	1	6	8	5	7	3	1		

13.6. ESCENARIO 6 (17 CLIENTES & 4 VEHÍCULOS)

La capacidad de los vehículos para este escenario es de 118 y la distancia máxima de recorrido de cada uno es de 143. El número mínimo de clientes a atender por cada vehículo debe ser superior a 3. En este escenario, debido a que el número de vehículos pasó a ser de 4, fue necesario elevar el parámetro “*Tiempo Máximo*” con el fin de asegurar una respuesta de calidad. “*Tiempo Máximo*” para este escenario fue de 2460 min, es decir 41 horas.

Tabla 22. Distancias entre nodos escenario 6

x	y
43	66
84	53

41	67
73	84
71	65
78	84
67	56
70	76
48	48
74	63
53	86
76	40
59	52
59	79
64	77
49	60
79	73
66	75

Fuente: Autor

Tabla 23. Demanda escenario 6

Demanda		
22	27	30
8	13	14
16	22	25
14	15	17
9	15	17
30	34	34
12	15	21
18	22	24
25	28	32
26	30	31
16	18	19
22	25	27
22	27	29
8	11	16
14	18	21
29	33	35
27	32	36

Tabla 24. Resultado escenario 6

Vehículos	Rutas							Distancia
1	1	16	13	12	17	8	1	383,6892891
2	1	14	15	18	11	1		
3	1	9	7	5	10	1		
4	1	3	4	6	2	1		

13.7. ESCENARIO 7 (20 CLIENTES & 4 VEHÍCULOS)

La capacidad de los vehículos para este escenario es de 700 y la distancia máxima de recorrido de cada uno es de 2550. El número mínimo de clientes a atender por cada vehículo debe ser superior a 4. En este escenario sucede lo mismo que en el escenario 6, de este modo el parámetro “*Tiempo Máximo*” en este caso fue de 4080 min, es decir 68 horas.

Tabla 25. Distancias entre nodos escenario 7

x	y
415	366
845	929
110	241
983	187
955	340
360	102
117	437
187	205
956	76
743	953
442	308
602	328
378	259
658	218
416	777
490	435

740	975
669	759
563	356
494	970
683	463

Tabla 26. Demandas Escenario 7

Demandas		
68	88	109
119	142	160
57	113	123
75	138	144
89	133	150
117	152	175
53	85	116
110	160	190
70	142	157
87	151	155
107	135	158
92	159	161
47	68	76
73	137	161
64	95	126
80	154	170
71	115	128
61	125	132
52	75	96
32	103	117

Tabla 27. Resultados Escenario 7

Vehículos	Rutas							Distancia
1	1	16	19	21	14	9	1	7522,87
2	1	12	18	17	20	15	1	
3	1	11	13	7	10	2	1	
4	1	6	8	3	5	4	1	

13.8. ANÁLISIS DE LAS INSTANCIAS

Observando los resultados obtenidos de cada instancia, se afirma que el modelo responder bastante bien a diferentes tipos de situaciones o problemas. Aun así, también se pudieron analizar ciertas limitantes que tendría este modelo en lo que se refiere a capacidad de memoria RAM y tiempo de procesamiento. Si bien el modelo usa diferentes estrategias para la generación de rutas individuales, si en algún punto la distancia máxima a recorrer por cada vehículo es muy grande, se generarían demasiadas rutas de este tipo, haciendo que el modelo arroje un error de memoria. Se estimó que dicho error ocurre cuando esta cantidad supera el valor de 37000.

Por otro lado, otra limitante del modelo se da en el segundo sector del mismo, el sector de combinación de rutas o generador de rutas globales, el cual, trabaja a partir de la cantidad de rutas individuales obtenidas mediante el proceso anterior en conjunto con el número de vehículos del problema. Se logró determinar que es necesario elevar el valor del parámetro “Tiempo máximo” entre más grande sea el modelo y más vehículos tenga, con el fin de obtener los mejores resultados posibles.

14. CONCLUSIONES

La aplicación de la matemática difusa en los diferentes problemas de programación lineal es una alternativa válida para dar manejo y solución a los mismos, y esto no solo se podría dar para el parámetro de demanda sino, adicionalmente, para otros tópicos de interés tales como la distancia máxima de recorrido o los tiempos de entrega, entre otros. Por otro lado también valdría la pena explorar esta temática en los modelos de programación enfocados a la parte de producción y distribución de planta.

La credibilidad dentro de la matemática difusa, resulta ser un factor que ayuda al manejo y aplicación de la lógica difusa en la programación lineal, dando un margen de flexibilidad a parámetros difusos. Esto eleva el número de soluciones factibles a los problemas que se estén tratando.

El diseño y creación de herramientas software debe ser un campo de constante estudio, esto debido a la cantidad de información que se puede adquirir de los fenómenos que pueden llegar a representar y a su vez al incremento de la productividad como consecuencia de la implementación de dichos algoritmos en la industria.

Al simular sistemas de ruteo de vehículos se observa que estos poseen una alta demanda de capacidad computacional, de este modo es necesaria la constante mejora de los algoritmos construidos y la construcción de nuevas propuestas de solución.

EL algoritmo PSO, como varios investigadores han afirmado con anterioridad, es una alternativa de bajo requerimiento computacional a la hora de buscar u optimizar resultados obtenidos mediante la aplicación de algoritmos matemáticos.

Es así como el estudio y aplicación de algoritmos heurísticos, además de la construcción de nuevas herramientas del mismo tipo, es un campo de estudio necesario que puede llegar en un futuro a ser los mecanismos necesarios para en todos los casos encontrar el resultado óptimo a estos complejos sistemas de ruteo.

15. RECOMENDACIONES

Se aconseja realizar un estudio del algoritmo propuesto, con el fin de buscar alternativas que ayuden a optimizar los procesos realizados en el sector de generación de rutas tanto individuales como globales de la solución propuesta. Dichas alternativas podrían ir desde la creación de bases de datos de rutas, hasta la implementación de otras heurísticas y metaheurísticas. Del mismo modo, es recomendable analizar el PSO y verificar si el hecho de que varias respuestas (partículas) se desplacen a la misma posición afecte en iteraciones posteriores la calidad de la respuesta dada.

Se deben construir bases de datos de los programas creados mediante proyectos de grado en la carrera de Ingeniería Industria, en el tema de programación lineal, con el fin que estas soluciones estén disponibles para los estudiantes que estén realizando su proyecto de grado en temas similares y sean usados como apoyo de tal modo se logren construir mejores herramientas software.

Algunos conceptos de la lógica difusa podrían llegar a ser una alternativa viable para la solución de otros modelos de programación lineal, es así como se hace necesario realizar un estudio acerca de otros problemas que se podrían llegar a tratar mediante parámetros difusos y credibilidad.

BIBLIOGRAFÍA

ALLAHVIRANLOO, Mahdiah; CHOW, Joseph. YJ. y RECKER, Will W. Selective vehicle routing problems under uncertainty without recourse. En: Transportation Research Part E: Logistics and Transportation Review. Febrero, 2014. vol. 62, p. 68-88

AVISO, Kathleen B.; TAN, Raymond R.; CULABA, Alvin B.; CRUZ JR, Jose B. Fuzzy input e output model for optimizing eco-industrial supply chains under water footprint constraints. En: Journal of Cleaner Production. Enero-Febrero, 2011. vol. 29, no. 2-3, p. 187-196.

CAGGIANI, Leonardo; OTTOMANELLI, Michele y SASSANELLI, Domenico. A fixed point approach to origin-destination matrices estimation using uncertain data and fuzzy programming on congested networks. En: Transportation Research Part C: Emerging Technologies. Marzo, 2013. vol. 28, p. 130-141

CANO, Yurilev Chalco. Algunas contribuciones a la teoría difusa multivariable. Febrero, 2004. Universidad Estatal de Campiñas. p. 1-10

DE OÑA, Juan; GÓMEZ, Penélope; MÉRIDA-CASERMEIRO, Enrique. Bilevel fuzzy optimization to pre-process traffic data to satisfy the law of flow conservation. En: Transportation Research Part C. Emerging Technologies. Febrero, 2011. vol. 19, no. 1, p. 29-39

ERBAO, Cao y MINGYONG, Lai. A hybrid differential evolution algorithm to vehicle routing problem with fuzzy demands. En: Journal of Computational and Applied Mathematics. Septiembre, 2009. vol. 231, no. 1, p. 302-310

ESHGHI, Kourosh y NEMATIAN, Javad. Special classes of fuzzy integer programming models with all-different constraints. En: Scientia Iranica. Enero-Junio 2009. vol. 16, no. 1, p. 1-10

GHAFFARI-NASAB, N.; GHAZANFAR AHARI, S. y GHAZANFARI, M. A hybrid simulated annealing based heuristic for solving the location-routing problem with fuzzy demands. En: Scientia Iranica. Junio, 2013. vol. 20, no. 3, p. 919-930

KENNEDY, James y EBERHART, Russell. Particle Swarm Optimization. Citado por: SALMAN, Ayed; AHMAD, Imtiaz y AL-MADANI, Sabah. Particle swarm optimization for task assignment problema. En: Microprocessors and Microsystems. Noviembre, 2002. vol. 26, no. 8, p. 363-371.

KENNEDY, James y EBERHART, Russell. Particle Swarm Optimization. En: Proceedings of the IEEE International Conference on Neural Networks. Noviembre-Diciembre, 1995. vol. 4, p. 1942–1948

KLIR, George J. y YUAN, Bo. Fuzzy sets and fuzzy logic: Theory and applications. Upper Saddle River: Prentice Hall Inc, 1995. p. 1-19

LIU, Chang-Shi y LAI, Ming-Yong. The vehicle routing problem with uncertain demand at nodes. En: Transportation Research Part E: Logistics and Transportation. Julio, 2009. vol. 45, no. 4, p. 517-524

NASRABADI, M.M., YAGHOUBI, M.A. y MASHINCHI, M. Separated continuous linear programs with fuzzy valued objective function. En: Scientia Iranica. Julio-Diciembre, 2010. vol. 17, no. 2, p. 105-118

PENG, Yang y QIAN, Ye-mei. A particle swarm optimization to vehicle routing problem with fuzzy demand. En: Convergence Information Tecnology. Agosto, 2010. Vol. 5, no. 6, p. 112-119

SELIM, Hasan; ARAZ, Ceyhun; OZKARAHAN, Irem. Collaborative production–distribution planning in supply chain: A fuzzy goal programming approach. En: Transportation Research Part E: Logistics and Transportation Review. Mayo, 2008. vol. 44, no. 3, p. 396–419

SHIVAIE, M., SEPASIAN, M.S. y SHEIKH-EL-ESLAMI, M.K. Multi-Objective transmission expansion planning using Fuzzy-Genetic Algorithm en Iranian Journal of Science and Technology. En: Transaction B: Engineering. Enero, 2011. vol. 35, no. 2, p. 141-159

TAN, Raymond R.; AVISO, Kathleen B.; BARILEA, Ivan U.; CULABA, Alvin B.; CRUZ JR, Jose B. A fuzzy multi-regional input–output optimization model for biomass production and trade under resource and footprint constraints. En: Applied Energy. Febrero, 2012. vol. 90, p. 154-160.

XU, Jiuping; YAN, Fang y LI, Steven. Vehicle routing optimization with soft time windows in a fuzzy random environment. En: Transportation Research Part E: Logistics and Transportation Review. Noviembre, 2011. vol. 47, no. 6, p. 1075-1091

ZHENG, Yongshuang y LIU, Baoding. Fuzzy vehicle routing model with credibility measure and its hybrid intelligent algorithm. En: Applied Mathematics and Computation. Mayo, 2006. vol. 176, no. 2, p. 673-683.

Anexo A. Programación del framework

```
%clc
clear all
close all

tic

%% Parametros iniciales
Pa=xlsread('Parametros.xlsx',1);
CC=xlsread('Parametros.xlsx',2);
DD=xlsread('Parametros.xlsx',3);

j_l=Pa(1,1);
i_l=Pa(2,1);
k_l=Pa(3,1);
L=Pa(4,1);
Cl=Pa(5,1);
Credi=Pa(6,1);
nr=Pa(7,1); %%rutas a guardar ordenadas
t_l=Pa(8,1);
Yp=Pa(9,1);
Yg=Pa(10,1);
w=Pa(11,1);
rdes=Pa(12,1);
t_iter=Pa(13,1);
t_iter=t_iter;

%% Parametros iniciales
fprintf('k=%d, i=%d \n',k_l,i_l)

%% Inicializacion de variables
```

```

C=CC(1:i_l,1:j_l);
D=DD(1:i_l,1:3);
d_l=length(D(:,1))-1;
dimension=2^(i_l*j_l)-1;
rutas=(2^(i_l))-1;
X=zeros(i_l,j_l);
Ypi=zeros(j_l,k_l);
R_Y=zeros(j_l,1);
R_X=zeros(i_l,j_l);
Min_t=zeros(9999999,1);
ca_x=zeros;
Min=zeros(nr,1);
Min_2=zeros(nr,1);
Min(1:nr,1)=9999;
vr=zeros(i_l,k_l);

Yr=zeros(j_l,k_l);
Xr=zeros(i_l,j_l);
aY1=zeros(j_l,k_l);
aX1=zeros(i_l,j_l);
aY2=zeros(j_l,k_l);
aX2=zeros(i_l,j_l);
aY3=zeros(j_l,k_l);
aX3=zeros(i_l,j_l);
aY4=zeros(j_l,k_l);
aX4=zeros(i_l,j_l);
aY5=zeros(j_l,k_l);
aX5=zeros(i_l,j_l);
aY6=zeros(j_l,k_l);

```

```

aX6=zeros(i_l,j_l);
aYru=zeros(j_l,k_l);
aXru=zeros(i_l,j_l);
aY7=zeros(j_l,k_l);
aX7=zeros(i_l,j_l);

aY12=zeros(j_l,k_l);
aX12=zeros(i_l,j_l);
aY42=zeros(j_l,k_l);
aX42=zeros(i_l,j_l);
Y_rt=zeros(j_l,k_l,k_l);
X_rt=zeros(i_l,j_l,k_l);
Y_ro=zeros(j_l,k_l,k_l,nr);
X_ro=zeros(i_l,j_l,k_l,nr);

Xl=zeros(1,i_l);
Xi=zeros(1,i_l);
Xi_rt=zeros(1,i_l,k_l);
Xi_ro=zeros(k_l,i_l,nr);
cli_rt=zeros(1,k_l);
cli_ro=zeros(1,k_l,nr);
cl_ro=zeros(1,k_l+1,nr);
Xi_cct=zeros(1,i_l-1,nr);
Xij=zeros(nr,i_l-1);
Xac=zeros(nr,i_l-1);
Xp=zeros(i_l-1,j_l-1);
Xpso=zeros(nr,i_l-1);
Xg=zeros(nr,i_l-1);

```

```
V=zeros(t_l,1);
fo=zeros(nr,t_l);
Xpi=zeros(k_l,i_l);
Xi_p=zeros(k_l,i_l);
Xi_r1=zeros(nr*2,i_l);
Xi_r2=zeros(nr*2,i_l);
Xi_2=zeros(k_l,i_l);
c_nr=0;
cont0=zeros(1,i_l);
cont1=0;
cont2=0;
cont3=0;
cont5=0;
cont6=0;
cont7=0;
cont12=0;
cont42=0;
cont8=0;

vr1=1;
vr2=1;
vr3=1;
vr4=1;
vr5=1;
vr6=1;
vr7=1;
vd=zeros(1,k_l);
num=zeros(1,i_l+1);
num(1)=1;
```

```
num(2)=1;
```

```
num(3)=1;
```

```
Y=zeros(i_l,1);
```

```
n_cj=2^(i_l-1)-1;
```

```
bits=i_l-1;
```

```
Y(1,1)=1;
```

```
Yc=zeros(bits,1);
```

```
for n=1:n_cj
```

```
    %% Combinaciones de Y
```

```
    %% se generan todas las posibles combinaciones de Y
```

```
    comb=single(dec2binvec(n,bits));
```

```
    Y(2:i_l,1)=comb;
```

```
    %% Credibilidad
```

```
    %% se aplica el criterio de credibilidad para descartar combinaciones que no  
    cumplen
```

```
    cre=zeros(4,1);
```

```
    %calcula todos los d1 d2 d3
```

```
    d=zeros(d_l,k_l);
```

```
    for q=1:3;
```

```
        for i=2:i_l
```

```
            d(q,1)=d(q,1)+D(i,q)*Y(i,1);
```

```
        end
```

```
    end
```

```

if Cl<=d(1,1)
    cre(1,1)=0;
end

if (Cl>=d(1,1) && (Cl<=d(2,1)))
    cre(2,1)=(Cl-d(1,1))/(2*(d(2,1)-d(1,1)));
end

if (Cl>=d(2,1) && (Cl<=d(3,1)))
    cre(3,1)=(-2*d(2,1)+d(3,1)+Cl)/(2*(d(3,1)-d(2,1)));
end

if Cl>=d(3,1)
    cre(4,1)=1;
end

vr7=1;
for c=1:4
    if cre(c,1)>=Credi
        vr7=0;
    end
end

Yc=zeros(bits,1);
%%Vecino mas cernano
%% se generan las rutas correspondientes a cada Y que cumplio
%%la condicion anterior utilizando la metaduristica del vecino mas cercano
if vr7<1
    c_yc=0;
    for i=1:i_l
        if Y(i,1)>0
            c_yc=c_yc+1;
        end
    end
end

```

```

        Yc(c_yc,1)=i;
    end
end

Ru=zeros(1,i_l+1);
Ru(1)=1;
tmp=0;

for q=1:c_yc-1
    qq=q-1;
    condicion=L;
    for d=2:c_yc-qq
        if C(Ru(q),Yc(d))<condicion
            condicion=C(Ru(q),Yc(d));
            tmp=d;
        end
    end
    Ru(q+1)=Yc(tmp);
    Yc(tmp:c_yc-1)=Yc(tmp+1:c_yc,1);
    Yc(c_yc-qq)=0;
end

Ru(c_yc+1)=1;
X=zeros(i_l,j_l);
for i=1:i_l
    if Ru(i+1)>0
        f=Ru(i);
        c=Ru(i+1);

```

```

        X(f,c)=1;
    end
end

%% restricción6
%%Se aplica la restricción 6 para eliminar las rutas que no cumplen
%%distancia máxima.
    c6=0;
    for i=1:i_l;
        for j=1:j_l;
            c6=c6+(X(i,j)*C(i,j));
        end
    end
%    c6=sum(sum(X.*C));

    if c6<=L
        ind=sum(Y(:,1))-1;
        cont0(ind)=cont0(ind)+1;

        Yr1(1:i_l,1,cont0(ind),ind)=Y(1:i_l,1);
        Xr1(1:i_l,1:j_l,cont0(ind),ind)=X(1:i_l,1:j_l);
        ca_x1(ind,cont0(ind))=ind;
        Xl1(1,1:(c_yc-1),cont0(ind),ind)=Ru(2:c_yc);
    end

    end %% end vr7 rutas por vecino mas cercano
end

d_max=max(ca_x1(:,1));

```

```

rdes1=rdes+1;
b=0;
a=1;
cont_ind=0;
ind_ca=zeros(1,i_l);
for i=rdes1:d_max
    cont_ind=cont_ind+1;
    b3=cont0(i);
    b=b+b3;
    ca_x(a:b)=ca_x1(i,1:b3);
    Xl(1,1:d_max,a:b)=Xl1(1,1:d_max,1:b3,i);
    Yr(1:i_l,1,a:b)=Yr1(1:i_l,1,1:b3,i);
    Xr(1:i_l,1:j_l,a:b)=Xr1(1:i_l,1:j_l,1:b3,i);
    ind_ca(1,cont_ind+rdes)=a;
    a=a+b3;
end

t_gr=toc;
fprintf('Generar rutas= %f \n\n',t_gr/60)
fprintf('%d | %d \n\n',sum(cont0),length(ca_x))

if d_max*k_l<i_l-1
    fprintf('las rutas generadas no cumplen con la demanda \n\n')
    break
end

if k_l==1 %%Para un solo vehiculo se realiza un proceso diferente ya que el
    %%ya que no se requiere combinar rutas ni se puede aplicar PSO

```

```

for fake=1:1
    X(1:i_l,1:j_l,1)=Xr(1:i_l,1:j_l,length(Xr(1,1,:)));
    Y(1:i_l,1,1)=Yr(1:i_l,1,length(Xr(1,1,:)));
    Xi(1,1:i_l)=Xl(1,1:i_l,length(Xr(1,1,:)));
    clientes(1)=ca_x(length(Xr(1,1,:)));
    %% restricción4
    %%se verifica que las rutas seleccionadas atiendan todos los clientes sin
    %%que llegue mas de un vehiculo a un punto.
    vr4=0;
    c4=zeros(j_l);
    for j=2:j_l;
        if vr4==0
            for i=1:i_l;
                for k=1:k_l;
                    c4(j)=c4(j)+X(i,j,k);
                end
            end
        end

        if c4(j)==1
            if j==j_l
                cont42=cont42+1;
                vr4=0;
            end
        else
            vr4=1;
        end
    end
end
end

```

```

%% Funcion objetivo
%%se aplica la funcion objetivo a los conjuntos de rutas que cumplieron
%%todas las condiciones anteriores
    if vr4==0
        cont8=cont8+1;
        for k=1:k_l
            for j=1:j_l;
                for i=1:i_l;
                    Min_t(cont8,1)=Min_t(cont8,1)+(X(i,j,k)*C(i,j));
                end
            end
        end
    end

    Min(1,1)=Min_t(cont8,1);
    Y_ro(1:j_l,1:k_l,1)=Y(1:j_l,1:k_l);
    X_ro(1:i_l,1:j_l,1:k_l,1)=X(1:i_l,1:j_l,1:k_l);
    Xi_ro(1:k_l,1:i_l,1)=Xi(1:k_l,1:i_l);
    cli_ro(1,,:,1)=clientes;

    end %end if vr4==0 funcion objetivo
end
t_t=toc;
fprintf('Total=%f \n\n',t_t/60)
fprintf('Min=%d \n',Min(1,1))
xlswrite('resultados.xls',Xi_ro(1:k_l,1:i_l,1),1);
xlswrite('resultados.xls',Min(1,1),2);
xlswrite('resultados.xls',Xi_ro(1:k_l,1:i_l,1),3);
xlswrite('resultados.xls',Min(1,1),4)

```

```

return
end %end if k_l==0 cuando solo hay un vehiculo.

```

```

t_iter=t_iter*60+t_gr;
nrutas=length(ca_x);
num2=ones(1,k_l);
num2(k_l+1)=0;
c1=i_l-1;
clientes=zeros(1,k_l);
t_limite=0;

```

```

%Restriciones que mezclan rutas.(K>1)

```

```

%%-----

```

```

for k=1:k_l
    num2(k)=(k_l-(k-1));
    Y(1:i_l,1,k)=Yr(1:i_l,1,(k_l-(k-1)));
    X(1:i_l,1:j_l,k)=Xr(1:i_l,1:j_l,(k_l-(k-1)));
    Xi(k,1:i_l)=XI(1,1:i_l,(k_l-(k-1)));
    clientes(k)=ca_x((k_l-(k-1)));
end
ofs=length(Yr(1,1,:))+1;
Yr(1:i_l,1:k_l,ofs:ofs+k_l)=zeros(i_l,k_l,k_l+1);
Xr(1:i_l,1:j_l,ofs:ofs+k_l)=zeros(i_l,j_l,k_l+1);
XI(1,1:i_l,ofs:ofs+k_l)=zeros(1,i_l,k_l+1);
ca_x(ofs:ofs+k_l)=zeros(1,k_l+1);
while num2(k_l+1)<1
    k=1;
    %num2

```

```

%%se generan todas las posibles combinaciones de las rutas existentes
%%para los k vehiculos a modo de obtener el conjunto total de rutas
%%necesarias para atender los clientes
while num2(k)>(nrutas-k)
    num2(k+1)=num2(k)+1;

    clientes(k+1)=ca_x(num2(k+1));
    Y(1:i_l,1,k+1)=Yr(1:i_l,1,num2(k+1));
    X(1:i_l,1:j_l,k+1)=Xr(1:i_l,1:j_l,num2(k+1));
    Xi(k+1,1:i_l)=XI(1,1:i_l,num2(k+1));

    for ik=k:-1:1
        num2(ik)=num2(ik+1)+1;
        Y(1:i_l,1,ik)=Yr(1:i_l,1,num2(ik));
        X(1:i_l,1:j_l,ik)=Xr(1:i_l,1:j_l,num2(ik));
        clientes(ik)=ca_x(num2(ik));
        Xi(ik,1:i_l)=XI(1,1:i_l,num2(ik));
    end
    k=k+1;
end

Y(1:i_l,1,1)=Yr(1:i_l,1,num2(1));
X(1:i_l,1:j_l,1)=Xr(1:i_l,1:j_l,num2(1));
clientes(1)=ca_x(num2(1));
Xi(1,1:i_l)=XI(1,1:i_l,num2(1));
num2(1)=num2(1)+1;

cont12=cont12+1;

```

```

    cls=sum(clientes);
    if (cls==c1)
%% restricción4
%%se verifica que las rutas seleccionadas atiendan todos los clientes sin
%%que llegue mas de un vehiculo a un punto.
        vr4=0;
        c4=zeros(j_l);
        for j=2:j_l;
            if vr4==0
                for i=1:i_l;
                    for k=1:k_l;
                        c4(j)=c4(j)+X(i,j,k);
                    end
                end
            end

            if c4(j)==1
                if j==j_l
                    cont42=cont42+1;
                    vr4=0;
                end
            else
                vr4=1;
            end
        end
    end
end
end
end

```

```

%% Funcion objetivo
%%se aplica la funcion objetivo a los conjuntos de rutas que cumplieron
%%todas las condiciones anteriores
if vr4==0
    cont8=cont8+1;
    t_limite=toc;
    for k=1:k_l
        for j=1:j_l;
            for i=1:i_l;
                Min_t(cont8,1)=Min_t(cont8,1)+(X(i,j,k)*C(i,j));
            end
        end
    end
end
%%para guardar todas las respuestas
%   Y_rt(1:i_l,1:k_l,cont8)=Y(1:i_l,1:k_l);
%   X_rt(1:i_l,1:j_l,1:k_l,cont8)=X(1:i_l,1:j_l,1:k_l);
%   Xi_rt(1:k_l,1:i_l,cont8)=Xi(1:k_l,1:i_l);
%   cli_rt(1,:,cont8)=clientes;

if Min_t(cont8,1)<=Min(1,1);
    for m=nr:-1:2
        Min(m,1)=Min((m-1),1);
        Y_ro(1:i_l,1:k_l,m)=Y_ro(1:i_l,1:k_l,(m-1));
        X_ro(1:i_l,1:j_l,1:k_l,m)=X_ro(1:i_l,1:j_l,1:k_l,(m-1));
        Xi_ro(1:k_l,1:i_l,m)=Xi_ro(1:k_l,1:i_l,(m-1));
        cli_ro(1,:,m)=cli_ro(1,:,m-1);
    end
    Min(1,1)=Min_t(cont8,1);

```

```

    Y_ro(1:j_l,1:k_l,1)=Y(1:j_l,1:k_l);
    X_ro(1:i_l,1:j_l,1:k_l,1)=X(1:i_l,1:j_l,1:k_l);
    Xi_ro(1:k_l,1:i_l,1)=Xi(1:k_l,1:i_l);
    cli_ro(1,.,1)=clientes;
else
    vr_ro=0;
    for n=2:nr
        if vr_ro<=0
            if Min_t(cont8,1)<=Min(n,1);
                for m=nr:-1:n
                    Min(m,1)=Min((m-1),1);
                    Y_ro(1:i_l,1:k_l,m)=Y_ro(1:i_l,1:k_l,(m-1));
                    X_ro(1:i_l,1:j_l,1:k_l,m)=X_ro(1:i_l,1:j_l,1:k_l,(m-1));
                    Xi_ro(1:k_l,1:i_l,m)=Xi_ro(1:k_l,1:i_l,(m-1));
                    cli_ro(1,.,m)=cli_ro(1,.,m-1);
                end
                Min(n,1)=Min_t(cont8,1);
                Y_ro(1:j_l,1:k_l,n)=Y(1:j_l,1:k_l);
                X_ro(1:i_l,1:j_l,1:k_l,n)=X(1:i_l,1:j_l,1:k_l);
                Xi_ro(1:k_l,1:i_l,n)=Xi(1:k_l,1:i_l);
                cli_ro(1,.,n)=clientes;
                vr_ro=1;
            end
        end
    end
end

end

end %end if vr4==0 funcion objetivo

```

```

elseif cls<c1
    cls1=sum(clientes(2:k_l));
    de_req=c1-cls1;
    if de_req<=d_max
        if ind_ca(1,de_req) >= num2(2)
            num2(1)=ind_ca(1,de_req);
        else
            num2(k_l+1)=1;
        end
    else
        num2(1)=nrutas;
    end
elseif cls>c1
    cls2=sum(clientes(k_l-1:k_l))+clientes(k_l-1);
    if cls2>c1
        if ca_x(1,num2(k_l))<ca_x(1,num2(k_l)+1)
            num2(k_l+1)=1;
        elseif clientes(k_l)*k_l>c1
            num2(k_l+1)=1;
        else
            num2(2)=nrutas-k;
            num2(1)=nrutas;
        end
    else
        num2(1)=nrutas;
    end
end%end if cl<=i_l cantidad de clientes

```

```

    if t_limite>t_iter
        num2(k_l+1)=1;
    end

end

t_cr=toc;
t_cr=t_cr-t_gr;
fprintf('Combinando rutas= %f \n\n',t_cr/60)

if cont8<nr
    nr=cont8;
end

%% PSO
%%se visualizan las rutas de forma lineal de la forma X[ | ]
for n=1:nr
    a=1;
    b=0;
    for k=1:k_l
        b1=cli_ro(1,k,n);
        b=b+b1;
        Xi_cct(1,a:b,n)=Xi_ro(k,1:cli_ro(1,k,n),n);
        a=a+b1;

        Xi_p(k,1,n)=1;
        Xi_p(k,2:(cli_ro(1,k,n)+1),n)=Xi_ro(k,1:cli_ro(1,k,n),n);
        Xi_p(k,cli_ro(1,k,n)+2,n)=1;
    end
end

```

```

end

v(1)=0;
Xmin=1;
Xmax=10;
%%Transformacion de la ruta a su version continua.
for i=1:nr
    for j=1:(i_l-1)
        Xij(i,j)=Xmin+((Xmax-Xmin)/(i_l-1))*(Xi_cct(1,j,i)-1+rand);
    end
end
Min_p=Min;
Xp=Xij;
Xac=Xij;
Xg=Xij(1, :, 1);
cli_xg=cli_ro(:, :, 1);
Min_g=Min(1);
X=Xij;
% X=zeros(nr,i_l-1,t_l);
%%inicio del PSO para las rutas seleccionadas
for t=2:t_l
    for i=1:(nr)
        %%hallando la velocidad y posicion.
        for j=1:(j_l-1)
            V(t)=w*V(t-1)+Yp*rand*(Xp(i,j)-Xac(i,j))+Yp*rand*(Xg(1,j)-Xac(i,j));
            X(i,j,t)=V(t)+X(i,j,t-1);
        end
    end
    % Xac(i,j)=X(i,j,t);

```

```

Xaux=X;
%%convertir a discreto la ruta resultante dle proceso anterior.
for j=1:(j_l-1)
    [M,IN]=min(Xaux(i,:,t));
    Xaux(i,IN,t)=9999999999;
    Xpso(i,IN,t)=j+1;
end
b=0;
a=1;
for k=1:k_l
    b3=cli_ro(1,k,i);
%    b3=cli_xg(1,k);
    b=b+b3;
    b1=b3+1;
    Xpi(k,1,i,t)=1;
    Xpi(k,2:b1,i,t)=Xpso(i,a:b,t);
    Xpi(k,b1+1,i,t)=1;
    a=a+b3;

    for j=1:b1
        a2=Xpi(k,j,i,t);
        b2=Xpi(k,j+1,i,t);
        fo(i,t)=fo(i,t)+C(a2,b2);
        Ypi(a2,k,i,t)=1;
    end
end

if fo(i,t)<=Min_p(i)
    %% criterio de credibilidad

```

```

%%se aplica la credibilidad a la ruta resultado para saber si
%%cumple las condiciones de credibilidad y distancia.
cre=zeros(4,1);
%calcula todos los d1 d2 d3
d=zeros(d_l,k_l);
vr7=0;
for kk=1:k_l;
    if vr7<1
        for q=1:3;
            for ii=2:i_l
                d(q,1)=d(q,1)+D(ii,q)*Ypi(ii,1);
            end
        end

        if Cl<=d(1,1)
            cre(1,1)=0;
        end

        if (Cl>=d(1,1)) && (Cl<=d(2,1))
            cre(2,1)=(Cl-d(1,1))/(2*(d(2,1)-d(1,1)));
        end

        if (Cl>=d(2,1)) && (Cl<=d(3,1))
            cre(3,1)=(-2*d(2,1)+d(3,1)+Cl)/(2*(d(3,1)-d(2,1)));
        end

        if Cl>=d(3,1)
            cre(4,1)=1;
        end

    vr7=1;
end

```

```

    for c=1:4
        if cre(c,1)>=Credi
            vr7=0;
            if k>=k_l
                Min_p(i)=fo(i,t);
                Xp(i,:)=X(i,:,t);
                Xi_p(:,:,i)=Xpi(:,:,i,t);
                if fo(i,t)<=Min_g
                    Min_g=fo(i,t);
                    X_g(1,:)=X(i,:,t);
                end
            end
        end
    end
end
end
end
end
end

end
end
Min_p2=Min_p;
for n=1:nr
    [M,IN]=min(Min_p2(:));
    Min_2(n,1)=Min_p2(IN);
    Xi_2(:,:,n)=Xi_p(:,:,IN);
    Min_p2(IN)=k_l*L;
end

t_t=toc;

```

```

t_pso=t_t-t_gr-t_cr;
fprintf('Aplicando PSO= %f \n\n',t_pso/60)
fprintf('Total=%f \n\n',t_t/60)
fprintf('Min=%d \n',Min(1,1))
fprintf('Min_pso=%d \n\n\n',Min_2(1,1))

a=1;
b=k_l;
for n=1:nr
Xi_r1(a:b,1:i_l)=Xi_2(1:k_l,1:i_l,n);
b=b+k_l;
a=a+k_l;
end
a=1;
b=k_l;
for n=1:nr
Xi_r2(a:b,1:i_l)=Xi_p(1:k_l,1:i_l,n);
b=b+k_l;
a=a+k_l;
end
    xlswrite('resultados.xls',Xi_r1,1);
    xlswrite('resultados.xls',Min_2,2);
    xlswrite('resultados.xls',Xi_r2,3);
    xlswrite('resultados.xls',Min,4)

```

Anexo B. Artículo.

UNA SOLUCIÓN PARA EL PROBLEMA DE RUTEO DE VEHÍCULOS CON DEMANDA DIFUSA MEDIANTE LA META-HEURÍSTICA DE OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS

A SOLUTION TO THE VEHICLE ROUTING PROBLEM WITH FUZZY DEMANDS BY META-HEURISTIC OF PARTICLE SWARM OPTIMIZATION

ROMÁN ALBERTO ÁLVAREZ CARDONA

Ingeniería industrial

Universidad Industrial de Santander

Roman.a.alvarez@gmail.com

HENRY LAMOS DÍAZ

Ph.D. en Física-Matemática

Hlamosdiaz@gmail.com

RESUMEN

En el presente documento, se expone un enfoque diferente a la resolución del problema de ruteo de vehículos con capacidad mediante la aplicación de la matemática difusa. Debido al problema que tiene las empresas a la hora de conseguir información válida en lo que se refiere a ciertos parámetros necesarios para la creación de las rutas a usar por sus respectiva flota de vehículos, es necesario explorar nuevas alternativas que logren brindar una solución viable y de calidad en búsqueda de minimizar los costos logísticos generados por el transporte de la mercancía. El concepto de credibilidad en la lógica difusa, puede usarse en conjunto con la programación lineal, como una alternativa válida para el manejo de aquellas situaciones en las cuales alguno de los parámetros de un problema no se conoce con certeza u obtenerlo no es rentable, y aun así, es necesario el embarque de una flota de vehículos que logre cumplir con la demanda de los clientes a un mínimo costo. De tal modo, el objetivo del presente documento es la creación de una herramienta de programación en Matlab®, enfocada a la solución del problema FVRP (*Fuzzy Vehicle Routing Problem – Problema de Ruteo de Vehículos Difuso*), en el cual la demanda será el factor que se tratara como un número difuso y por consiguiente bajo el concepto de credibilidad. Sin embargo para la creación de dicha herramienta también se darán conceptos básicos de la teoría difusa y una breve explicación de las heurísticas y Metaheurísticas aplicadas al modelo.

PALABRAS CLAVE: Problema de ruteo de vehículos con parámetros difusos, Problema de ruteo de vehículos con capacidad, PSO, Enjambre de partículas, Matemática difusa.

ABSTRAT

In this paper, a different problem solving vehicle routing capacity by applying fuzzy mathematical approach outlined. Due to the problem that businesses in getting valid information in regard to certain parameters needed to create paths to use for their respective fleet of vehicles, it is necessary to explore new alternatives that achieve provide a viable and Quality seeking to minimize logistics costs generated by the transport of

goods. The concept of credibility on fuzzy logic can be used in conjunction with linear programming as a valid alternative for handling situations in which some of the parameters problem is not known with certainty or obtain it is not profitable, and yet, it is necessary boarding a fleet of vehicles which would meet the customer demand at a minimal cost. Thus, the objective of this document is to create a programming tool in Matlab®, focused on solving the problem FVRP (*Fuzzy Vehicle Routing Problem*), in which the demand will be the factor that is treated as a fuzzy number and therefore the concept of credibility. However for creating such a tool basics of fuzzy theory and a brief explanation of the heuristics and Metaheuristics applied to the model were also given.

KEY WORDS: Vehicle routing problem with fuzzy parameters, vehicle routing problem with capacity, PSO, particle swarm optimization, fuzzy mathematics.

1. INTRODUCCIÓN

La investigación de operaciones es una rama de las matemáticas que trata sobre el desarrollo de modelos de optimización en búsqueda de soluciones prácticas y eficientes para problemas en la industria. El problema VRP (Vehicle Routing Problem - Problema de Ruteo de Vehículos) desde hace ya algún tiempo es objeto de estudio por ingenieros e investigadores, debido a la gran cantidad de variantes que existen de este tipo de problemas y la complejidad de ellos.

El CVRP es uno de los ejemplos más sencillos de los problemas de ruteo, el cual consiste en satisfacer las demandas de un conjunto de clientes distribuidos en una zona geográfica por una flota de vehículos con una capacidad definida (Q), que inician y finalizan su recorrido en un sitio determinado, el cual es llamado deposito. Sin embargo, en este tipo de problema la demanda del cliente entra a tomar importancia, a causa de que no todas las empresas logran tener los datos exactos de la demanda a tiempo (antes de construir las rutas a utilizar). Todo esto puede llegar a afectar a la organización elevando los costos de transporte debido al uso de un sistema de rutas “no óptimas” para el problema.

Peng y Qian (2010) exponen un modelo para el FVRP cuya demanda es manejada como un valor difuso. A su vez, este modelo es resuelto por los mismos utilizando la metaheurística de Enjambre de Partículas (PSO). Este modelo puede dar un manejo a diferentes problemas que se evidencian en el día a día en las industrias y a nuestro alrededor; como la recolección de basuras, el reabastecimiento de cajeros automáticos o estaciones de gasolina, la distribución de productos tienda a tienda, la entrega de correos,

etc. Es de este modo que se deben construir herramientas software enfocadas a la solución de este tipo de problema, con el fin implementar apropiadamente en el mundo de la industria estos conocimientos.

2. JUSTIFICACIÓN

El problema del transporte de carga ocasiona grandes costos a las empresas comparados con el costo logístico total, en conjunto con un mercado competitivo el cual exige entregas de pedidos a tiempo de manera eficiente, hacen de éste problema una necesidad de estudio para investigadores y organizaciones. Es así como la implementación de modelos de programación lineal enfocados hacia la optimización de las rutas y por tanto la minimización de los costos en este campo es de vital importancia, así como la construcción e implementación de herramientas informáticas que faciliten su uso en la industria.

El presente artículo, plantea una solución factible para el FVRP mediante la construcción de una herramienta software práctica para su uso en el mundo empresarial.

3. CVRP Y FVRP

El CVRP es uno de los problemas básicos dentro de este campo y por tanto existen diferentes variantes para dicho problema. A continuación, se expondrá el modelo en su forma clásica, para posteriormente modificarlo a su contraparte difusa:

F. O.

$$= \sum_i^n \sum_j^n \sum_k^p C_{ij} X_{ijk} \quad (1)$$

S.A.

$$\sum_i^n X_{ijk} = Y_{jk} \quad j = 0, 1, 2, \dots, n; k = 1, 2, \dots, p. \quad (2)$$

$$\sum_j^n X_{ijk} = Y_{ik} \quad i = 0, 1, 2, \dots, n; k = 1, 2, \dots, p. \quad (3)$$

Esta restricción evita que un vehículo realice el mismo recorrido más de una vez:

$$\sum_i^n X_{ijk} = \sum_i^n X_{jik} \quad j = 0, 1, 2, \dots, n; k = 1, 2, \dots, p. \quad (4)$$

La siguiente restricción, es la restricción de capacidad y crea los conjuntos de clientes que pueden ser atendidos en un solo recorrido por el vehículo k :

$$\sum_i^n d_i Y_{ik} \leq CI \quad k = 1, 2, \dots, p. \quad (5)$$

La restricción presentada a continuación garantiza que cada cliente sea visitado una vez:

$$\sum_i^n \sum_k^p X_{ijk} = 1 \quad j = 0, 1, 2, \dots, n. \quad (6)$$

Restricción para que cada vehículo salga y regrese a la bodega:

$$\sum_j^n X_{0jk} = 1 \quad k = 1, 2, \dots, p. \quad (7)$$

Restricción que determina la máxima distancia recorrida posible para cada vehículo:

$$\sum_i^n \sum_j^n C_{ij} X_{ijk} \leq L \quad k = 1, 2, \dots, p. \quad (8)$$

Esta última restricción da las dimensiones del modelo:

$$X, Y \in [0, 1], i, j = 0, 1, 2, \dots, n, k = 1, 2, \dots, p \quad (9)$$

El objetivo del CVRP es encontrar las mejores rutas posibles para una flota de vehículos cuya capacidad inicial CI es un valor fijo conocido.

El FVRP que se tratara en este documento es básicamente el CVRP, pero el valor de la demanda será un número difuso de forma triangular.

3.1. FVRP

Los números difusos triangulares son conjuntos difusos cuya función de pertenencia se caracteriza por estar conformada por dos ecuaciones de recta que se interceptan en un solo punto, este punto es el que posee la máxima amplitud dentro de dicho conjunto y por tanto el nivel de pertenencia de este punto es de 1 para un conjunto $A(x)$.

$$A(x) = \begin{cases} 1 & \text{Para } x \in [a, b] \\ L(x) & \text{Para } x \in (-\infty, a) \\ A(x) & \text{Para } x \in (b, \infty) \end{cases} \quad (10)$$

Ejemplo:

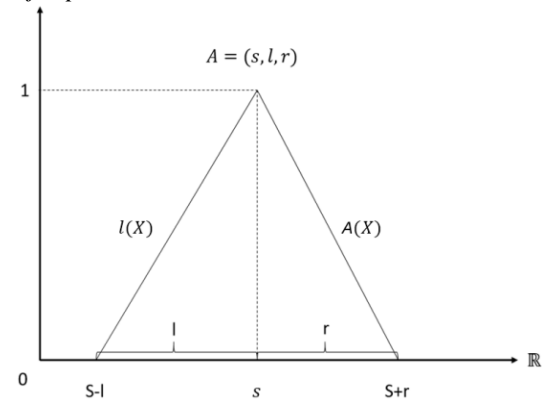


Figura 120: Número difuso triangular $A(x)$.

Debido a las características especiales de este tipo de conjunto difuso, estos en especial pueden ser representados mediante tres números reales ($S-L, S, S+R$), los cuales representan los extremos y el punto con mayor grado de pertenencia de dicho conjunto.

3.2. Una solución al problema FVRP

Debido a que el único parámetro que se tratara como un valor difuso es la demanda, la única restricción que se verá afectada es la de capacidad, de este modo mediante la teoría difusa se le podría dar una solución al modelo FVRP, en donde la restricción que lleva el parámetro difuso d_i podría escribirse del siguiente modo:

Sea d_i un número difuso triangular el cual representa la demanda difusa de i clientes, entonces,

$$d_i = d_{i1}, d_{i2}, d_{i3} \quad (11)$$

Por tanto la restricción de capacidad del CVRP se podría escribir de la siguiente manera:

$$\sum_i^n \text{Max}(d_{i1}, d_{i2}, d_{i3})Y_{ik} \leq CI \quad (12)$$

$k = 1, 2, \dots, p.$

Sin embargo, esta solución se basa en la transformación del modelo FVRP en un CVRP clásico, dado que se asumiría la demanda difusa como un número real representado por el valor más alto dentro de dicho conjunto.

3.3. Credibilidad

Conceptos básicos de la teoría difusa introducidos por Zheng y Liu (2006), afirman que un número difuso puede ser representado mediante sus ecuaciones de Necesidad, Posibilidad y Credibilidad. La credibilidad, da una alternativa para dar manejo al parámetro difuso de la credibilidad sin la necesidad de resolver el modelo de la forma vista anteriormente, De este modo es necesario ver la función de credibilidad para un número difuso:

Sea V un número difuso triangular de la forma $V = (v_1, v_2, v_3)$, entonces, se tiene que su función de credibilidad para $v \geq q$ esta descrita del siguiente modo:

$$= \begin{cases} 1 & \text{si } q \leq v_1 \\ \frac{2v_2 - v_1 - q}{2(v_2 - v_1)} & \text{si } v_1 \leq q \leq v_2 \\ \frac{v_3 - q}{2(v_3 - v_2)} & \text{si } v_2 \leq q \leq v_3 \\ 0 & \text{si } q \geq v_3 \end{cases} \quad (13)$$

Haciendo un análisis general del modelo CVRP, se asume que la restricción de capacidad podría manejarse bajo el criterio de credibilidad del siguiente modo:

$$\text{Cre} \left(\sum_i^n d_i Y_{ik} \leq CI \right) \geq \text{Cre}^* \quad (14)$$

$k = 1, 2, \dots, p.$

Lo que afirma esta ecuación, es que la credibilidad de que un vehículo k pueda cumplir con una ruta dada debe ser mayor a una credibilidad dada por el experimentador Cre^* . Este parámetro de credibilidad simboliza el índice de preferencia del despachador, es decir, representa la actitud del distribuidor frente al riesgo de poder o no poder cumplir con la demanda del cliente siguiente. Dicho valor, como se ha mencionado con antelación, es dado por el experto, puesto que es necesario tener especial cuidado con éste parámetro. Si la relación $\text{Cre} \geq \text{Cre}^*$ se cumple, el vehículo k en ruta se dirigirá al siguiente cliente; de lo contrario el vehículo se dirigirá al depósito terminando con su recorrido y otro vehículo será quien cumpla con la siguiente demanda. De este modo, se puede decir que, el valor de Cre^* influye directamente en las rutas de los vehículos y en la distancia total recorrida por cada uno durante su trayecto.

4. HEURÍSTICAS Y METAHEURÍSTICAS PARA LA OPTIMIZACIÓN DEL MODELO

Debido a que el CVRP es un problema que pertenece a la clase *NP hard* (*Non-deterministic Polynomial-time hard*), esto significa que la solución para situaciones en la que el número de clientes es relativamente grande, del mismo modo sucede con el FVRP planteado. De este modo es necesario hacer uso de heurísticas y metaheurísticas que minimicen lo más posible este problema, dado que estos últimos son algoritmos con bajo costo computacional.

4.1. Heurística del vecino más cercano

El vecino más cercano es una metodología básica para dar solución a los problemas de transporte dentro de la rama de programación lineal y optimización. Aunque las soluciones obtenidas comúnmente no son óptimas, esta heurística es una buena opción para la generación de conjuntos de soluciones factibles.

Para la aplicación de esta heurística, en un principio se parte de un punto origen y se observa cual es el nodo más próximo, de ahí se traza la ruta hasta él, luego, ubicado en el nodo se observa cual es el más cercano de los que faltan por visitar y se traza la ruta al siguiente nodo, este proceso se repite una y otra vez hasta haber visitado todos los nodos y por ultimo regresa al origen. Se puede ver con claridad que esta heurística en la mayoría de los casos no da una respuesta muy óptima; sin embargo, para realizar el análisis sobre las rutas factibles y la aplicación de otras heurísticas y/o metaheurísticas para optimizar la respuesta, resulta ser una metodología bastante simple y eficaz.

4.2. Optimización por enjambre de partículas (PSO)

El PSO es un algoritmo metaheurístico enfocado a la optimización de resultados, es comúnmente utilizado en diversas líneas de la Investigación de operaciones y se encuentra fundamentado en el comportamiento de los enjambres de las abejas. El PSO fue propuesto en 1995 por los investigadores Kennedy y Eberhart (1995), cuyo objetivo era el estudio de la colaboración en el desplazamiento de los organismos (aves, abejas, peces etc.), sin embargo durante los últimos años dicho algoritmo se ha logrado implantar exitosamente en la optimización de modelos de programación lineal y otras ramas de la ingeniería.

El objetivo de la optimización por enjambre de partículas (nube de partículas) es la determinación de la mejor posición que posea un organismo en una población, mediante el estudio de la ubicación y velocidad de aquellos que lo rodean, de este modo, análogamente en la investigación de operaciones se busca la mejor respuesta posible mediante el estudio de un conjunto de soluciones

factibles denominadas partículas, las cuales poseen su propia posición y velocidad en un universo U , y por medio de ecuaciones matemáticas son trasladadas de un lugar a otro en el interior de este conjunto universo con el fin de ubicar aquella solución que cumpla en mayor medida las exigencias del experimentador.

El PSO maneja dos ecuaciones una de posición de la partícula y otra de su velocidad (Salman, et al, 2002):

- Ecuación de Posición:

$$X_t = X_{t-1} + V_t \quad (15)$$

- Ecuación de Velocidad:

$$V_t = (W * V_{t-1}) + ((Y_p * Ram) * (X_p - X_{ij})) + ((Y_g * Ram) * (X_g - X_{ij})) \quad (16)$$

La posición X_t simboliza la ubicación de la partícula en un tiempo t , V_t simboliza la velocidad de la partícula en un tiempo t , Y_p y Y_g son denominados grados de proximidad y según su valor determinaran la dirección de la velocidad de la partícula, X_p y X_g son la mejor posición local conocida y la mejor posición global conocida respectivamente, x_{ij} es la posición en cada partícula, Ram simboliza un número aleatorio que pertenece al conjunto $[0,1]$ y w es una constante de equilibrio.

Para el uso de las ecuaciones del PSO es necesario en algunos casos trasladar los datos de un espacio discreto a uno continuo, a esto se le denomina codificar los datos, en estos casos se utiliza la ecuación que se enuncia a continuación, donde x_{max} y x_{min} son el rango de valores, y_i es el número a modificar y n es la cantidad de números del conjunto.

- Ecuación de discreto a continuo:

$$X_{ij} = X_{min} + (X_{max} - X_{min}) * \frac{(Y_i - 1 + Ram)}{n} \quad (17)$$

En un momento será necesario decodificar para obtener la ruta que genera el PSO, de este modo,

para decodificar el conjunto lo único que hay que hacer es organizar los datos de menor a mayor, y colocar el dato de menor valor al número más pequeño en el conjunto original y el segundo menor al siguiente más pequeño, y así sucesivamente hasta que se decodifique completamente el número.

5. PAUTAS PARA CONSTRUCCIÓN DEL FRAMEWORK EN MATLAB

El framework construido está basado en el modelo descrito en la sección 3 de este documento, en donde la respuesta se verá optimizada mediante el uso del PSO y las rutas generadas que este necesita serán construidas mediante el vecino más cercano.

A continuación se exponen los sectores de la programación (ver Figura 2) y las pautas que se tomaron en cuenta para su elaboración:



Figura 2. Sectores del algoritmo programado.

5.1. Generación de sub-rutas

Debido a que el PSO necesita de la generación de un conjunto de resultados factibles (partículas), se utiliza la restricción de capacidad, que para el caso también es la restricción de credibilidad, con el fin de generar conjuntos de clientes que el vehículo K se crea pueda visitar y cumplir con su demanda en un solo recorrido sin que se exceda la capacidad del mismo durante su trayecto. Posteriormente se aplica la heurística del Vecino más cercano para construir a partir de los conjuntos de clientes sub-rutas viables para el proceso es decir, los conjuntos de clientes pasan a ser conjuntos de

rutas para un solo vehículo. En este punto, se aplica a estas rutas la restricción de distancia máxima, con el fin de eliminar aquellas rutas que tengan un desplazamiento muy largo y que un vehículo no logre hacer en un solo recorrido.

Ahora, se decide implantar un parámetro el cual se denomina “Mínimo número de clientes a atender”, el objetivo de la introducción de este parámetro es que el experimentador pueda decidir cuál va a ser el número mínimo de nodos que un solo vehículo pueda visitar, de este modo eliminar algunas rutas creadas por los pasos anteriores y que no son necesarias para el modelo.

Un inconveniente de este sector del modelo es la incapacidad de la memoria del computador para almacenar grandes cantidades de rutas; debido a éste inconveniente se debe tener cuidado al ingresar al modelo los parámetros generales, dado que si se generan gran cantidad de rutas individuales es posible que el modelo genere un error de memoria.

5.2. Generación de rutas globales

Una vez completados los pasos anteriores, se tiene un conjunto de rutas las cuales pueden darse para un solo vehículo, con el cual se dispone a construir el conjunto de rutas globales para la aplicación del PSO. Las rutas obtenidas son de este tipo $[2, 5, 3]$, significa que algún vehículo podrá ir de 1 a 2, de 2 a 5 y de 5 a 3 para luego regresar al origen (1), esta sería la ruta de tan solo un vehículo. En el caso que se tuvieran dos vehículos, para construir la ruta global, es necesario buscar en el conjunto de rutas aquellas o aquella que se complementen y logren cumplir con la demanda de los n clientes, es decir, si se toma la ruta $[2, 5, 3]$ y son sólo 5 clientes $(2, 3, 4, 5, 6)$, entonces deberá haber unas rutas que contengan los clientes restantes, un ejemplo sería $[4, 6]$, por consiguiente una ruta global podría verse desde esta forma $[2, 5, 3 | 4, 6]$, de este modo se generan las rutas globales requeridas por el experimentador para posteriormente aplicar el PSO.

Un inconveniente en este paso es el tiempo de ejecución, debido a que se deben realizar una gran cantidad de iteraciones en la búsqueda de las rutas que cumplan con los requisitos y poder almacenar

las mejores respuestas. Para reducirlo se implementan estrategias que eviten iteraciones innecesarias; a pesar de esto, el tiempo no mejora de manera considerable, así que se opta por implementar un parámetro de tiempo “*Tiempo máximo*”. Dicho parámetro establece un tiempo máximo de generación de rutas globales de forma que cuando se utilice el modelo, se pueda detener este proceso en algún punto y utilizar las mejores rutas que se hayan logrado obtener hasta el momento.

Se podría decir que el parámetro “*Tiempo máximo*” decide en gran medida la calidad de las rutas globales que se utilizaran posteriormente en el PSO, esto debido a que la programación lo que busca es encontrar las mejores n rutas: entre más tiempo el modelo tenga para generar las rutas globales es muy probable que la calidad de éstas sea mucho mejor.

5.3. Aplicación del PSO

El tercer sector del modelo, aplica la heurística PSO al conjunto de rutas obtenido por medio de los dos pasos anteriores. Una consideración que se tiene en este sector, es mantener la posición de las partículas durante las interacciones, aunque estas posiciones no se traducen en rutas factibles para el modelo. Sin embargo, las rutas que se mantengan guardadas serán las mejores posiciones locales entre las cuales deberá estar la mejor global, y solo serán alteradas si en alguna iteración surge una posición para la partícula factible que sea más óptima.

Cada uno de los sectores del modelo pueden ser identificados fácilmente dentro del mismo, ya que se hizo un escrito referente a su función antes de cada uno, además de estos sectores, se pueden también observar otras funciones que hacen diferentes tipos de programaciones.

6. MODO DE USO DEL FRAMEWORK

6.1. Parámetros

Los parámetros generales del modelo se ingresan al mismo por medio de un archivo de MS Excel bajo el nombre de “Parametros”, ubicado en el mismo directorio que el archivo de Matlab®. Los

parámetros se pueden modificar fácilmente y se dividen en las tres hojas del mismo archivo.

En la primera hoja de cálculo se pueden modificar todos los parámetros que involucran las dimensiones del modelo a tratar, en primer lugar se encuentra la cantidad de nodos que va a tener el modelo que se identifican con las letras “ j_L ” y “ i_L ”, refiriéndose a la longitud de “ j ” y de “ i ” respectivamente. El parámetro “ k_L ” indica el número de vehículos que tendrá el modelo. “ L ” simboliza la longitud máxima a recorrer de cada vehículo y “ CI ” la capacidad inicial de los mismos. “*Credibilidad*”, como su nombre lo indica, es la credibilidad con la cual se van a construir las rutas.

El parámetro “ nr ” es el mejor número n de rutas encontradas por el algoritmo antes de aplicar el PSO y las cuales se guardan mediante el paso 2 del mismo modelo. “ t ” es el número de iteraciones que realizará el PSO antes de almacenar los resultados. También se encuentran “ Yp ”, “ Yg ” y “ w ”: los parámetros de las fórmulas utilizadas en el PSO.

Adicionalmente, “*rutas a descartar*” es un parámetro que elimina del número de rutas iniciales todas aquellas que para el concepto del experimentador, son rutas que cumplen con un número de clientes lo suficientemente pequeño para no ser tomadas en cuenta dentro del desarrollo del modelo.

	A	B
1	j_L	18
2	i_L	18
3	k_L	3
4	L	275
5	CI	160
6	<i>Credibilidad</i>	0,3
7	nr	80
8	t	750
9	Yp	0,2
10	Yg	1,5
11	w	0,1
12	<i>rutas a descartar</i>	4
13	<i>tiempo máximo [m]</i>	600
14		
15		
16		

Imagen 1. Parámetros generales hoja 1.

Por último, “*tiempo máximo*” es un parámetro que se agregó al sector de combinación de rutas, con

el fin de dar un límite de tiempo a este proceso, dado que los tiempos de combinación en ciertos casos son demasiado elevados si se realizaran todas las iteraciones posibles. Debido a esto y a que el PSO no requiere de una gran cantidad de rutas sino solo de algunas de ellas, se decidió incluir dicho parámetro.

En la hoja 2 se ingresa la matriz de distancias entre nodos, Sin embargo es necesario mencionar que las distancias de ir de un nodo a ese mismo, se toman como valores extremadamente altos para el modelo, esto con el fin de que al correr el algoritmo este no tome estas rutas como factibles y las descarte.

En la tercera y última hoja del documento MS Excel, se ingresan las demandas difusas triangulares, teniendo en cuenta que el nodo uno es el origen de los recorridos o bodega y por tanto la demanda de este es de 0.

6.2. Resultados

Los resultados del modelo también son almacenados en un archivo de MS Excel, el cual es creado por el mismo modelo y tiene el nombre de “Resultados”.

En las hojas 3 y 4 del archivo se ubican las rutas y el resultado al evaluarlas en la función objetivo, de aquellas encontradas mediante el generador de rutas globales, es decir, estas son las partículas usadas en el PSO. Del mismo modo en las hojas 1 y 2 encontramos las rutas y el resultado al evaluarlas en la función objetivo de los resultados finales del modelo, refiriéndose a las rutas ya optimizadas por el PSO.

Las rutas resultado no salen agrupadas según el número de vehículos, por tanto es necesario agruparlas, este es algo sencillo dado que salen en orden.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
4	1	11	16	14	18	6	8	12	1	0	0	0	0	0	0	0	0	0
5	1	9	10	15	13	17	1	0	0	0	0	0	0	0	0	0	0	0
6	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0
7	1	11	16	14	18	6	8	12	1	0	0	0	0	0	0	0	0	0
8	1	9	10	15	13	17	1	0	0	0	0	0	0	0	0	0	0	0
9	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0
10	1	11	16	14	18	6	8	12	1	0	0	0	0	0	0	0	0	0
11	1	9	10	17	13	15	1	0	0	0	0	0	0	0	0	0	0	0
12	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0
13	1	11	16	14	18	6	8	1	0	0	0	0	0	0	0	0	0	0
14	1	12	9	10	15	13	17	1	0	0	0	0	0	0	0	0	0	0
15	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0
16	1	11	16	14	18	6	8	1	0	0	0	0	0	0	0	0	0	0
17	1	12	9	10	15	13	17	1	0	0	0	0	0	0	0	0	0	0
18	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0
19	1	11	16	14	18	6	8	1	0	0	0	0	0	0	0	0	0	0
20	1	12	9	10	15	13	17	1	0	0	0	0	0	0	0	0	0	0
21	1	4	5	7	2	3	1	0	0	0	0	0	0	0	0	0	0	0

Imagen 2. Rutas resultado del algoritmo.

7. DISEÑO DE EXPERIMENTOS DEL MODELO PROGRAMADO

Para verificar el correcto funcionamiento del modelo y obtener un mejor manejo de él, se decide realizar un estudio del modelo mediante un diseño de experimentos; de este modo es necesario obtener conocimiento de que parámetros impactan de mayor forma la variable respuesta y a su vez determinar qué valor usar para optimizar dicho resultado. El diseño de experimentos para evaluar el modelo es un diseño 2^{k-p} , en el cual se realizaron 8 corridas experimentales. Los computadores utilizados en este proceso son Intel Core i7, además la versión de Matlab® usada es la 2014^a.

En la Tabla 1, observamos los factores que se tiene en cuenta para el diseño de experimentos a realizar, además se pueden ver cómo fueron manejados durante el proceso. Por otra parte la Tabla 2 muestra los datos de entrada del experimento, es decir, los resultados de cada una de las 8 corridas realizadas.

Tabla 1. Factores del diseño experimental

	+	-
Credibilidad	0,9	0,3
W	0,9	0,1
Yg	1,5	0,3
Tg (Tiempo máximo)	1440	60

Tabla 2. Datos de entrada para el diseño experimental.

564,080868
629,320951
659,009372
734,918619
615,477198
578,613504
646,851834
598,200641

En base a los resultados obtenidos, y observando cada una de las gráficas obtenidas mediante el software Minitab 17, en las que se pueden ver los impactos de los parámetros *TG*, *w*, *Cre* y *Yg*, además de las interacciones entre estos, se tiene que:

Los factores “*w*” y “*Tg*” (Tiempo máximo) además de la interacción de “*Credibilidad*” y “*w*”, son los factores e interacción que más impacto tienen sobre la variable respuesta en dicho algoritmo (ver Figura 3).

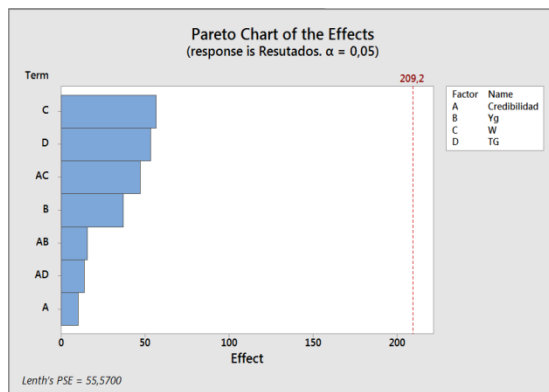


Figura 3. Diagrama de Pareto para los efectos

El valor de la credibilidad aunque no tiene impacto que los demás factores, si se realiza un análisis más profundo, sí debe tener un impacto considerable; esto se debe a que al observar el número de rutas generadas con un valor alto de credibilidad y las generadas con un valor bajo la diferencia no es mucha, en parte debido a que estos resultados son optimizados por el PSO. No obstante, se puede deducir con facilidad que al tener valores bajos de credibilidad los vehículos

podrían llegar a visitar más clientes por ruta que con valores altos de dicho parámetro.



Figura 4. Grafica de efectos para cada factor.

Por medio de la Figura 4, se puede deducir que los efectos “*w*”, “*Yg*” y “*Credibilidad*”, son factores que e deben de manejar en bajo, lo contrario pasa con “*Tg*” (Tiempo máximo), el cual como es lógico, debe de manejarse en alto. Lo anterior debe de ser así, si se desean construir rutas con bajos costos, no obstante, antes de asumir el valor de la credibilidad bajo, se debe pensar que esto incrementara el riesgo de que un vehículo llegue a un cliente y no pueda atender su demanda, elevando los costos logísticos de transporte.

Tabla 3. Contraste de resultados

Resultados antes del diseño de experimentos	Resultado después del diseño de experimentos
564,080868	541,8136093
629,320951	
659,009372	
734,918619	
615,477198	
578,613504	
646,851834	
598,200641	

El factor “*Tg*” es un parámetro que si bien el modelo dice que es necesario manejarlo en alto, se recomendaría manejar el tiempo lo más elevado posible, no simplemente 24 horas, dado que las rutas generadas serán mejores entre más tiempo tenga el modelo para realizar este proceso.

8. VERIFICACIÓN DEL MODELO MEDIANTE INSTANCIAS

Como se ha podido ver, el modelo programado ofrece una nueva alternativa de solución a

sistemas de ruteo. Sin embargo, este debe de responder correctamente a diferentes tipos de problemas que se puedan llegar a dar durante su uso práctico en el mercado. De este modo, se debe verificar su comportamiento frente a diferentes tipos de situaciones (instancias), para observar los resultados arrojados por dicho modelo frente a cada una de estas y verificar su funcionalidad.

En la Figura 5, se ve cómo se tomaron los parámetros generales para cada uno de los escenarios planteados. Los parámetros faltantes son diferentes para cada escenario dado. Por otra parte, se decidió tomar el parámetro de credibilidad de 0.8, esto con el fin de disminuir el riesgo, de tal modo se minimicen el número de situaciones en las que en realidad el vehículo llegue al cliente y no logre cumplir con su demanda.

Se plantearon 7 escenarios en donde, el número de clientes y vehículos para cada escenario se ven en la Tabla 4.

j ₁			
i ₁			
k ₁			
L			
C ₁			
Credibilidad	0,8		
nr	80	Numero de rutas a guardar	
t	500	Numero de interacciones PSO	
Y _p	0,2		
Y _g	0,3		
w	0,1		
rutas a descartar			
tiempo maximo [m]	1440		

Figura 5. Parámetros generales de los escenarios

Tabla 4. Escenarios.

Escenario	Clientes	Vehículos
1	8	1
2	8	2
3	14	2
4	14	3
5	17	3
6	17	4
7	20	4

Tabla 5. Resultados de instancias

Escenario	Resultado
1	283

2	690
3	2847
4	1479
5	4105
6	383
7	7522

Al realizar cada una de las corridas para cada escenario, se observó que el modelo responde correctamente para cauno de estos, sin embargo es necesario aclarar que los el parámetro “Tiempo máximo” en los dos últimos escenarios se elevó a 3 días, es decir fue necesario que el modelo dedicara por lo menos 4320 minutos, de tal modo se pudiese obtener un resultado. Los resultados de cada una de las instancias pueden verse en la Tabla 5.

Mediante este proceso se pudo analizar una limitante que tendría este modelo en lo que se refiere a capacidad de memoria RAM. Si bien el modelo usa diferentes estrategias para la generación de rutas individuales, si en algún punto la distancia máxima a recorrer por cada vehículo es muy grande, se generarían demasiadas rutas de este tipo, haciendo que el modelo arroje un error de memoria. Se estimó que dicho error ocurre cuando esta cantidad supera el valor de 37000.

9. CONCLUSIÓN

La aplicación de la matemática difusa en los diferentes problemas de programación lineal, es una alternativa válida para dar manejo y solución a los mismos y esto no solo se podría dar para el parámetro de demanda, sino para otros sectores de interés tales como la distancia máxima de recorrido o los tiempos de entrega entre otros, Por otro lado también valdría la pena explorar esta temática en los modelo de programación enfocados a la parte de producción y distribución de planta.

La credibilidad dentro de la matemática difusa, resulta ser un factor que ayuda al manejo y aplicación de la lógica difusa en la programación lineal, dando un margen de flexibilidad a parámetros difusos elevando el número de soluciones factibles a los problemas que se estén tratando.

El diseño y creación de herramientas software debe ser un campo de constante estudio, esto debido a la cantidad de información que se puede adquirir de los fenómenos que pueden llegar a representar y a su vez al incremento de la productividad como consecuencia de la implementación de dichos algoritmos en la industria.

Al simular sistemas de ruteo de vehículos, se observa que estos poseen una alta demanda de capacidad computacional, de este modo es necesaria la constante mejora de los algoritmos construidos y la construcción de nuevas propuestas de solución.

EL algoritmo PSO como varios investigadores han afirmado con anterioridad, es una alternativa de bajo requerimiento computacional a la hora de buscar u optimizar resultados obtenidos mediante la aplicación de algoritmos matemáticos. Es así como el estudio y aplicación de algoritmos heurísticos además construcción de nuevas herramientas del mismo tipo, es un campo de estudio necesario que puede llegar en un futuro a ser los mecanismos necesarios para en todos los casos encontrar el resultado óptimo a estos complejos sistemas de ruteo.

10. REFERENCIAS

PENG, Yang y QIAN, Ye-mei. A particle swarm optimization to vehicle routing problem with fuzzy demand. En: Convergence Information Technology. Agosto, 2010. vol. 5, no. 6, p. 112-119

ZHENG, Yongshuang y LIU, Baoding. Fuzzy vehicle routing model with credibility measure and its hybrid intelligent algorithm. En: Applied Mathematics and Computation. Mayo, 2006. vol. 176, no. 2, p. 673-683

KENNEDY, James y EBERHART, Russell. Particle Swarm Optimization. En: Proceedings of the IEEE International Conference on Neural Networks. Noviembre-Diciembre, 1995. vol. 4, p. 1942-1948

KENNEDY, James y EBERHART, Russell. Particle Swarm Optimization. Citado por:

SALMAN, Ayed; AHMAD, Imtiaz y AL-MADANI, Sabah. Particle swarm optimization for task assignment problema. En: Microprocessors and Microsystems. Noviembre, 2002. vol. 26, no. 8, p. 363-371.