EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación y Desarrollo Tecnológico

Tesis de Doctorado

Desarrollo de nuevas heurísticas para incrementar la eficiencia del algoritmo Fuzzy C-Means

presentado por

MC. Carlos Fernando Moreno Calderón

como requisito para la obtención del grado de

Doctor en Ciencias de la Computación

Directora de tesis

Dra. Alicia Martínez Rebollar

Co-director de tesis

Dr. Eddie Helbert Clemente Torres

Cuernavaca, Morelos, México. Agosto de 2026

cenidet
Centro Nacional de Investigación y Desarrollo Tecnológico

cenidet® Centro Nacional de Investigación y Desarrollo Tecnológico	ACEPTACIÓN DE IMPRESIÓN DEL DOCUMENTO DE TESIS DOCTORAL	Código: CENIDET-AC-006-D20
		Revisión: O
	Referencia a la Norma ISO 9001:2008 7.1, 7.2.1, 7.5.1, 7.6, 8.1, 8.2.4	Página 1 de 1

Cuernavaca, Mor., a 12 de junio de 2026

DR. CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO
P R E S E N T E

AT'n: DRA. ANDREA MAGADÁN SALAZAR
PRESIDENTA DEL CLAUSTRO DOCTORAL DEL DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

Los abajo firmantes, miembros del Comité Tutorial del estudiante **M.C. CARLOS FERNANDO MORENO CALDERÓN** manifiestan que después de haber revisado el documento de tesis titulado **"Desarrollo de nuevas heuristicas para incrementar la eficiencia del algoritmo Fuzzy C-Means"**, realizado bajo la dirección de la **Dra. Alicia Martínez Rebollar** y codirigido por el **Dr. Eddie Helbert Clemente Torres,** el trabajo se ACEPTA para proceder a su impresión.

A T E N T A M E N T E

Excelencia en Educación Tecnológica®
"Conocimiento y Tecnología al Servicio de México"

DRA. ALICIA MARTÍNEZ REBOLLAR
TECNM/CENIDET

DR. EDDIE HELBERT CLEMENTE TORRES
TECNM/CENIDET

DR. DANTE MÚJICA VARGAS
TECNM/CENIDET

DR. HUGO ESTRADA ESQUIVEL
TECNM/CENIDET

DR. JAVIER ORTIZ HERNÁNDEZ
TECNM/CENIDET

DRA. ADRIANA MEXICANO SANTOYO
TECNM/INSTITUTO TECNOLÓGICO DE CIUDAD VICTORIA

c.c.p: C Verónica Sotelo Boyas/ Jefa del Departamento de Servicios Escolares
c.c.p: C Nóe Alejandro Castro Sánchez / Jefe del Departamento de Ciencias Computacionales
c.c.p: Expediente

Centro Nacional de Investigación y Desarrollo Tecnológico
Subdirección Académica

Cuernavaca Mor, 22/junio/2026
Oficio No. SAC/178/2026
Asunto: Autorización de impresión de tesis

CARLOS FERNANDO MORENO CALDERÓN
CANDIDATO AL GRADO DE DOCTOR EN CIENCIAS
DE LA COMPUTACIÓN
P R E S E N T E

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **"Desarrollo de nuevas heurísticas para incrementar la eficiencia del algoritmo Fuzzy c-Means"**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
"Conocimiento y Tecnología al servicio de México"

EDUCACIÓN | TECNOLÓGICO NACIONAL DE MÉXICO
CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
SUBDIRECCIÓN ACADÉMICA

CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO

c.c.p. Departamento de Ciencias Computacionales
Departamento de Servicios Escolares

CMAZ/lmz

cenidet
Centro Nacional de Investigación y Desarrollo Tecnológico
Interior Internado Palmira S/N, Col. Palmira,
C.P. 62490, Cuernavaca, Morelos Tel. 01 (777) 3627770, ext. 4106,
e-mail: acad_cenidet@tecnm.mx | cenidet.tecnm.mx

Dedicatoria

Con cariño dedico este trabajo a:

Dios, por darme fuerzas cada día, guiar mis pasos
y permitirme cumplir una meta más en la vida.

A mis padres, Carlos Moreno y Erika Calderón,
por los esfuerzos que realizan cada día por mí,
por las enseñanzas que me han brindado,
por sus cansancios y desvelos,
por los regaños y el amor que me tienen;
Dios les pague todo lo que han hecho por mí.
¡Los amo!

A mis hermanos, Juan, Yoselin, Helem, Jaaziel y José,
por las risas, las travesuras, las aventuras y peleas,
he aprendido mucho con ustedes, Dios les pague.

Agradecimientos

Mi más profundo agradecimiento a mi directora de tesis Dra. Alicia Martínez Rebollar y a mi codirector de tesis Dr. Eddie Helbert Clemente Torres, por brindarme el apoyo, la guía y confianza en esta etapa de mi formación profesional. Al Dr. Joaquín Pérez Ortega y a la Dra. Sandra Silvia Roblero Aguilar gracias por el apoyo y la motivación constante en el inicio y culminación de mi formación.

A los miembros de mi comité tutorial: Dr. Dante Mújica Vargas, Dr. Hugo Estrada Esquivel, Dr. Javier Ortiz Hernández y Dra. Adriana Mexicano Santoyo por sus consejos y observaciones realizadas en el desarrollo de este trabajo de investigación. Al personal administrativo, gracias por el apoyo para que los trámites que realicé durante mi estancia fueran fáciles.

A la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) y al Tecnológico Nacional de México / Centro Nacional de Investigación y Desarrollo Tecnológico (TecNM/CENIDET), por la oportunidad y facilidad brindadas para la realización de esta tesis.

A Lupita, gracias por tu paciencia, tu compañía, y el amor que me has brindado.

A mi familia, que a pesar de la distancia siempre preguntan ¿cómo estoy? Gracias por su apoyo. A mis mejores amigos: Arnulfo Carrillo, Wilson Alexis y Cesar Gilberto por el apoyo brindado, los consejos y sobre todo las carcajadas.

Al grupo de Seminario de Investigación, gracias por sus comentarios, su apoyo y la convivencia. A mis amigos y compañeros de laboratorio, han hecho que el tiempo que pasé aquí sea una gran experiencia.

¡Muchas gracias!

Resumen

En esta tesis se desarrolla investigación de frontera en las áreas de Ciencia de Datos y Reconocimiento de Patrones. En particular, el problema que se aborda consiste en reducir la complejidad computacional del algoritmo de agrupamiento difuso denominado Fuzzy *C*-Means. Dicho algoritmo se ha aplicado con éxito en la solución de conjuntos de datos en dominios como la salud, visión por computadora, segmentación de imágenes, entre otros. Sin embargo, la complejidad computacional del algoritmo es muy alta, del orden $O(nc^2di)$. Dicha complejidad limita severamente la aplicación del algoritmo en conjuntos de datos grandes.

Como propuesta de solución se desarrollaron dos nuevas heurísticas que probaron reducir la complejidad del algoritmo. La primera heurística denominada P-FCM, introduce un nuevo criterio de paro del algoritmo, el cual consiste en determinar la diferencia porcentual del valor de la función objetivo en dos iteraciones consecutivas, y en caso que dicha diferencia sea menor a un umbral dado el algoritmo para. La segunda heurística denominada HPFCM, es híbrida, e integra por una parte un mecanismo para optimizar los centroides iniciales del algoritmo y por otra integra la primera heurística.

La validación de las heurísticas propuestas se realizó de manera experimental. Se diseñó un conjunto de experimentos en los que se resolvieron conjuntos de datos reales y sintéticos. Con base en los resultados experimentales se observó que las heurísticas reducen la complejidad computacional. En el mejor de los casos las heurísticas logran reducir hasta un 98.17% el número total de iteraciones respecto al algoritmo Fuzzy *C*-Means, además se alcanzó una mejora de la calidad de solución de hasta 39.73%. Con base a estos resultados, se logró reducir la complejidad computacional, obteniendo una mejora en la eficiencia computacional del algoritmo Fuzzy *C*-Means. Esto permite que el proceso de agrupación sea más rápido y los resultados que espera un usuario sean iguales o mejores que los resultados que se esperaría por el algoritmo Fuzzy *C*-Means.

Finalmente, es destacable que la principal contribución de esta investigación, es la reducción del tiempo de solución, manteniendo los mismos recursos computacionales. Sin duda, dicha contribución impactará de manera significativa en conjuntos de datos grades, como las que comúnmente se presentan en problemas reales.

Abstrac

This thesis presents cutting-edge research in the fields of Data Science and Pattern Recognition. Specifically, it addresses the problem of reducing the computational complexity of the fuzzy clustering algorithm known as Fuzzy *C*-Means. This algorithm has been successfully applied to the analysis of datasets in fields such as healthcare, computer vision and image segmentation, amongst others. However, the computational complexity of the algorithm is very high, of the order $O(nc^2di)$. This complexity severely limits the application of the algorithm to large datasets.

As a proposed solution, two new heuristics were developed which were found to reduce the algorithm's complexity. The first heuristic, known as P-FCM, introduces a new criterion for terminating the algorithm, which involves calculating the percentage difference in the value of the objective function between two consecutive iterations; if this difference is less than a given threshold, the algorithm converges. The second heuristic, known as HPFCM, is a hybrid approach; it incorporates, on the one hand, a mechanism for optimize the algorithm's initial centroids and, on the other, the first heuristic.

The proposed heuristics were validated experimentally. A series of experiments was designed in which real and synthetic datasets were analyzed. Based on the experimental results, it was observed that the heuristics reduce computational complexity. In the best-case scenario, the heuristics managed to reduce the total number of iterations by up to 98.17% compared to the Fuzzy *C*-Means algorithm; furthermore, an improvement in solution quality of up to 39.73% was achieved. Based on these results, it was possible to reduce computational complexity, thereby improving the computational efficiency of the Fuzzy C-Means algorithm. This enables the clustering process to be faster, and the results a user expects are the same as, or better than, those that would be expected from the Fuzzy C-Means algorithm.

Finally, it is worth noting that the main contribution of this research is the reduction in solution time, whilst maintaining the same computational resources. This contribution will undoubtedly have a significant impact on large datasets, such as those commonly encountered in real-world problems.

Contenido

Pág.

Lista de Tablas

Pág.

Lista de Figuras

Pág.

Capítulo 1

1. Introducción

El presente capítulo describe la investigación doctoral ubicada en el área de algoritmos de agrupamiento de objetos. Se presenta la motivación, la descripción general del planteamiento del problema, se describe el objetivo general y se detallan los objetivos específicos, y finalmente se presenta la organización general del documento de tesis.

1.1 Motivación

El objetivo del agrupamiento de objetos consiste en particionar un conjunto de datos que poseen varios atributos numéricos en subconjuntos, de tal manera que los objetos que pertenecen a un grupo son similares entre ellos y distintos a los objetos que pertenecen a otros grupos.

Los algoritmos para el agrupamiento de datos se han utilizado en diferentes disciplinas. Algunas de ellas son: reconocimiento de patrones, segmentación de imágenes, minería de datos, entre otros (Yang, M. S., 1993; Nayak, J. et al, 2014). Sin embargo, una de las limitaciones para el agrupamiento de datos es su costo computacional.

Algunas publicaciones especializadas mencionan que los volúmenes de datos crecen constantemente (Shen, Y. et al, 2020). Es por ello que el análisis de agrupamiento es ampliamente usado para poder comprender los volúmenes generados por el mundo real. Ya que con ello se puede tener una interpretación de cómo se relacionan los datos entre sí.

Para el agrupamiento de datos se han desarrollado varios algoritmos, entre ellos K-Means (MacQueen, J., 1967) y Fuzzy *C*-Means (Bezdek, J. C., 1981). La principal característica de K-Means es que genera particiones duras, esto quiere decir que los objetos

pertenecerán a un solo grupo, mientras que Fuzzy *C*-Means realiza una partición difusa, donde los objetos tendrán un grado de pertenencia a cada grupo. Por esta razón, una de las ventajas de Fuzzy *C*-Means sobre K-Means es la calidad de resultados, cuando la representación de los grupos tiene una interpretación matizada, lo cual está relacionado con la partición difusa. Sin embargo, la complejidad computacional de Fuzzy *C*-Means es superior a la de K-Means.

1.2 Planteamiento del problema

El problema que se aborda en esta investigación consiste en desarrollar una variante del algoritmo Fuzzy *C*-Means, cuyo tiempo de ejecución sea menor que el de Fuzzy *C*-Means estándar y sin que se reduzca significativamente la calidad de la solución.

La complejidad temporal del algoritmo Fuzzy *C*-Means es $O(ndc^2i)$ y la complejidad de K-Means es $O(ndci)$ (Ghosh, S. y Kumar, S., 2013) donde:

- n = número de objetos,
- d = número de atributos,
- c = número de centroides o grupos,
- i = número de iteraciones.

Si bien Fuzzy *C*-Means ha mostrado mejores resultados en calidad que otros algoritmos de agrupamiento, como K-Means, su tiempo de ejecución es una limitación importante a medida que crece el tamaño de las instancias a resolver.

Para dar una idea de la complejidad de Fuzzy *C*-Means, en la Tabla 1 y en la Figura 1 se muestran ejemplos de la solución de una instancia, resuelta con diferentes valores de números de grupos, con K-Means y Fuzzy *C*-Means para n = 1000, d =15 y i= 20. Nótese que, en este ejemplo, solamente se ha cambiado una de las variables de la expresión de complejidad. Obsérvese como a medida que aumenta el número de c grupos la complejidad de Fuzzy *C*-Means se incrementa.

Tabla 1. Contraste entre los valores de complejidad de K-Means y Fuzzy C-Means

No. grupos	Complejidad temporal K-Means	Complejidad temporal Fuzzy C-Means
2	600,000	1,200,000
3	900,000	2,700,000
4	1,200,000	4,800,000
5	1,500,000	7,500,000
6	1,800,000	10,800,000

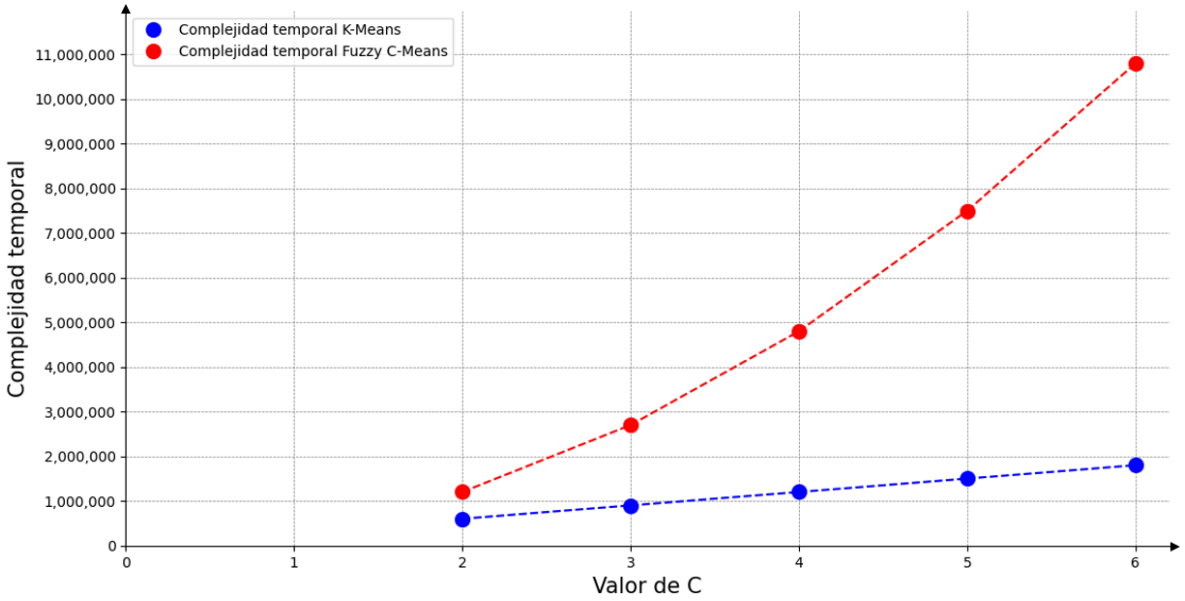


Figura 1. Comparación de la complejidad temporal entre K-Means y Fuzzy C-Means.

El reto principal de esta investigación consiste en reducir la complejidad de Fuzzy *C*-Means estándar. Reduciendo el valor de una, o varias, de las variables que intervienen en la expresión de complejidad. Lo anterior, sin afectar significativamente la calidad de la solución del algoritmo, esto es, sin que los valores de la función objetivo de la variante de Fuzzy *C*-Means difiera significativamente de los de Fuzzy *C*-Means estándar.

En la Figura 2, en el eje *x* se representa el tamaño de las instancias que se deben resolver; en el eje *y* el valor de la complejidad temporal. La línea roja representa la propuesta de esta investigación, la cual consiste en reducir la complejidad temporal de Fuzzy *C*-Means. Es importante aclarar que la diferencia de la línea roja con respecto a la complejidad temporal de Fuzzy *C*-Means (línea verde), solamente es ilustrativa, con la intención de mostrar el concepto de la reducción de complejidad.

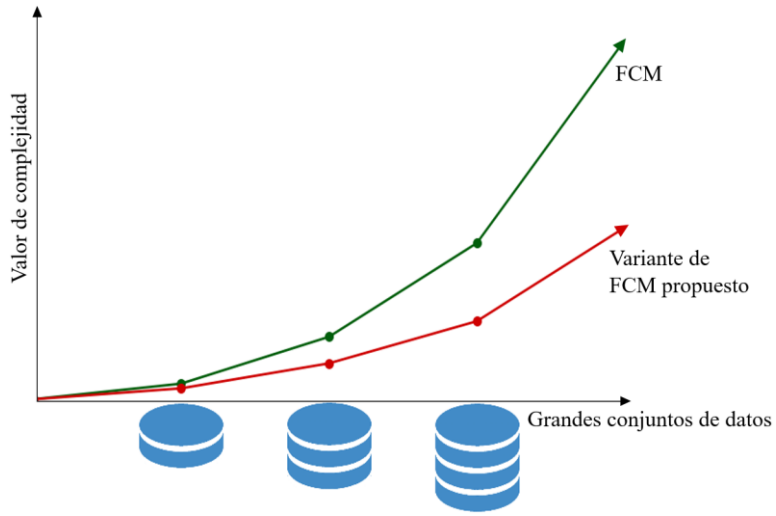


Figura 2. Descripción gráfica del problema.

Uno de los estudios con mayor reconocimiento en la comunidad científica (Garey, M. y Johnson, D., 1979) ha demostrado que los problemas de agrupamiento como el que resuelve Fuzzy *C*-Means tienen complejidad NP-*hard*. Para estos, actualmente no existen algoritmos eficientes que los resuelvan en tiempo polinomial. Tales problemas son, por lo tanto, intratables computacionalmente. Lo anterior, confirma que la mejora del algoritmo de agrupamiento Fuzzy *C*-Means sea un problema abierto de investigación, relevante y con vigencia actual.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar nuevas heurísticas para mejorar la eficiencia del algoritmo Fuzzy *C*-Means en la solución instancias.

1.3.2 Objetivos específicos

Derivados del objetivo general, se proponen los siguientes objetivos particulares:

1. Utilizar conjuntos de datos reales o sintéticos para identificar patrones de solución en la observación del comportamiento de Fuzzy *C*-Means.
2. Desarrollar dos heurísticas que intervengan en las fases de inicialización y convergencia del algoritmo Fuzzy *C*-Means para mejorar la eficiencia temporal en la solución instancias.

3. Validar experimentalmente las heurísticas propuestas para incrementar la eficiencia del algoritmo Fuzzy *C*-Means, usando conjuntos de datos reales o sintéticos generados exprofeso.

1.4 Organización del documento

El presente documento se organiza de la siguiente manera: en el Capítulo 2, se presenta el marco teórico donde se definen conceptos clave para la comprensión de esta investigación; en el Capítulo 3, se presenta el estado del arte que incluye antecedentes y los trabajos relacionados a este trabajo de investigación; en el Capítulo 4, se presenta la propuesta de solución, la cual se enfoca a las fases de inicialización y convergencia del algoritmo Fuzzy *C*-Means; en el Capítulo 5, se muestra la experimentación realizada y el análisis de resultados; en el Capítulo 6, se presentan las conclusiones de la presente investigación, los trabajos futuros y los logros obtenidos.

Capítulo 2

2. Marco Teórico

En este capítulo, se presentan de forma sintetizada definiciones relacionadas al campo de los algoritmos de agrupamiento, las cuales son fundamentales para la comprensión de esta investigación.

2.1 Aprendizaje supervisado y no supervisado

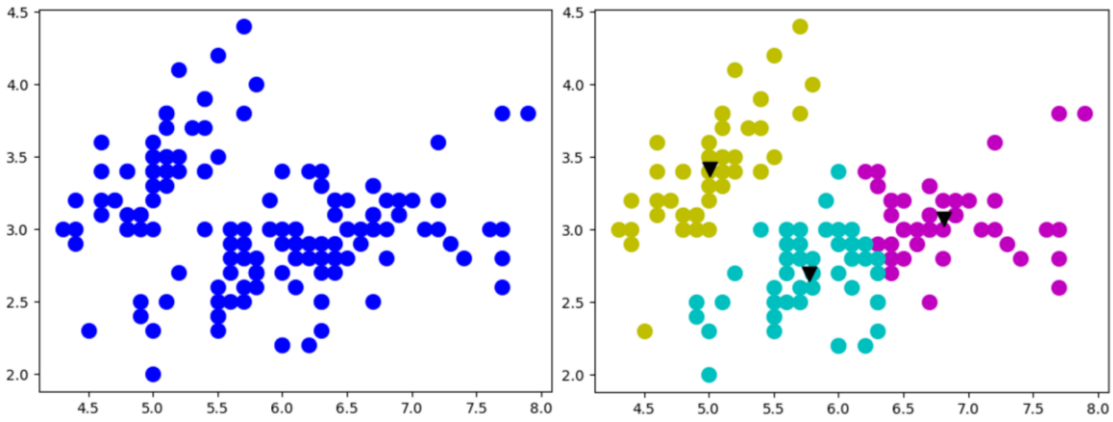
El aprendizaje supervisado es una rama de *machine learning*, donde un conjunto de datos etiquetados facilita la predicción de salidas (Segovia, N y Guzmán, A, 2025). Los modelos de aprendizaje supervisado trabajan con conjuntos de datos de *n* objetos, donde cada objeto consiste en *d* atributos y cada objeto tiene una variable etiqueta. Estos modelos predicen una salida con base a las características de los datos de entrada. Los modelos de aprendizaje supervisados más comunes son: Redes neuronales artificiales (McCulloch, W.S. y Pitts, W. 1943), Máquinas de soporte vectorial (Vapnik, V. 1995), K-vecinos más cercanos (Cover, T y Hard, P. 1967), entre otros.

El aprendizaje no supervisado, es una rama de *machine learning*, donde un conjunto de datos no etiquetados es usado para encontrar patrones de solución sin una instrucción explicita sobre la salida (Segovia, N. y Guzmán, A. 2025). Estos modelos en lugar de predecir, se basan en los atributos de los datos para encontrar características que los distingan y darles una interpretación.

2.1.1 Agrupamiento

El agrupamiento o *clustering* es una técnica usada en distintas áreas como estadística, minería de datos y la ciencia de datos. El objetivo del *clustering* consiste en agrupar o clasificar un

conjunto de datos en subconjuntos, de tal manera que los datos de un grupo comparten características similares entre sí y características diferentes a los datos que pertenecen a otros grupos (Ezugwu, A. et al., 2022).



a) Conjunto de datos b) Conjunto de datos agrupado

Figura 3. Agrupamiento del conjunto de datos Iris en dos dimensiones

Una característica importante en el proceso de *clustering*, es el uso de una métrica de similitud. Se sabe que una norma de distancia es una técnica para determinar la distancia entre los puntos de datos, de entre las cuales destacan la Euclidiana, Mahalanobis y Manhattan (Xu, R. y Wunsch, D., 2005).

Existen diferentes tipos de agrupamiento, algunos de ellos son de tipo partición, jerárquico, basados en densidad, entre otros (Mahid, M. et al., 2021). El agrupamiento de partición, divide un conjunto de ***n*** objetos en ***k*** grupos, el cual puede ser asignado por el usuario o determinado por algún pre procesamiento de datos; el agrupamiento jerárquico, parte de un único grupo y van generando divisiones sobre esos grupos, de tal manera que los datos formarán parte de una jerarquía particional; los basados en densidad, identifican grupos o regiones densamente pobladas por puntos de datos, sin embargo, discrimina puntos aislados de los grupos como ruido.

2.1.2 Redes Neuronales Artificiales

Las redes neuronales artificiales (RNA), son modelos computacionales diseñadas para imitar ciertas capacidades del cerebro, con el fin de realizar el reconocimiento de patrones y aprendizaje automático (McCulloch, W.S. y Pitts, W. 1943). Una RNA está compuesta por

nodos interconectados que modelan las neuronas del cerebro, y aristas que realizan la sinapsis, imitando el funcionamiento del cerebro y su capacidad de aprender (Rosenblatt, F. 1958). En la Figura 4, se muestra una representación de una red neuronal artificial.

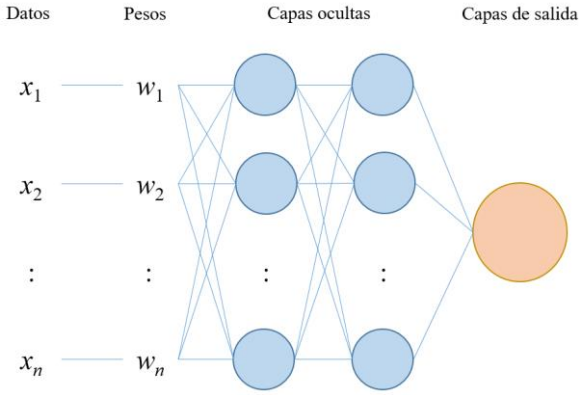


Figura 4. Representación de una red neuronal artificial.

Los componentes de las RNA son tres:

1. **Neuronas:** La arquitectura de una RNA incluye capas de entrada, donde se reciben los datos; una o varias capas ocultas, donde se realiza el procesamiento de los datos; y una capa de salida, donde se muestra el resultado.
2. **Pesos:** Los pesos son ponderaciones asignadas entre las neuronas, que determinan la importancia de cada característica en el resultado final.
3. **Funciones de activación:** La función de activación, es una suma ponderada de las entradas transformándolas a una salida no línea.

2.2 Fuzzy *C*-Means

El algoritmo Fuzzy *C*-Means (Bezdek, J. et al., 1984) es un algoritmo de agrupamiento particional de tipo difuso. A diferencia del algoritmo K-Means (MacQueen, J., 1967), que realiza particiones en donde un objeto pertenece a un solo grupo, Fuzzy *C*-Means usa una función de membresía para medir la pertenencia que tiene cada objeto a los grupos. El proceso asigna a los objetos valores entre 0 y 1, y se incorpora un factor difuso que determina que tan difuminados están los grupos entre sí.

El algoritmo Fuzzy *C*-Means está fundamentado en la teoría de conjuntos difusos, el cual fue propuesto por (Zadeh, L., 1965). En su publicación define que, dado un conjunto de datos X con un elemento genérico de X denotado por x, entonces $X = \{x\}$, y un conjunto

difuso *A* en el conjunto *X* se caracteriza por una función de membresía $f_A(x)$ que asocia con cada punto en *X* un número real en el intervalo [0, 1], con el valor de $f_A(x)$ en *x* representando el grado de membresía de *x* en *A*. De este modo, el valor más cercano a $f_A(x)$ a la unidad, el grado de membresía de *x* en *A* será mayor.

El concepto de agrupamiento difuso fue introducido por (Ruspini, E., 1969), fundamentado en la teoría de conjuntos difusos desarrollada por Zadeh en su publicación (Zadeh, L. 1965). En (Dunn, J., 1974) se extiende el término de agrupamiento difuso a partir del concepto de la partición dura, tomando como referencia el algoritmo K-Means, donde define la función objetivo de Fuzzy *C*-Means, como se muestra en la Expresión 1.

$$J_D(U,V) = \sum_{j=1}^{n}\sum_{i=1}^{c} \mu_{ij}^2 \, ||x_j - v_i||^2 \tag{1}$$

En (Bezdek, J. C., 1981) se define un cambio en la Expresión 1, al exponente de ponderación se le puede asignar cualquier valor mayor a uno. Además, el autor sugiere que los mejores valores para asignarle a *m* son $1.5 \leq m \leq 3$.

Finalmente, la función objetivo de Fuzzy *C*-Means se describe en la Expresión 2. Sea $X = \{x_1, \dots, x_n\}$ el conjunto de datos de *n* objetos a particionar, donde $x_i \in \mathcal{R}^d$ para $i = 1, \dots, n$, y *c* es el número de grupos donde $2 \leq c \leq n$.

$$J_m(U,V) = \sum_{i=1}^{n}\sum_{j=1}^{c} \mu_{ij}^m \left\|x_i - v_j\right\|_2^2 \tag{2}$$

Donde $U = \mu_{ij}$ es el grado de pertenencia de cada objeto *i* a cada grupo *j*; $V = \{v_1, \dots, v_c\}$ es el conjunto de centroides, donde v_j es el centroide del grupo *j*; *m* es el exponente de ponderación o factor difuso, que altera el grado de pertenencia, $m > 1$ y $\left\|x_i - v_j\right\|_2^2$ indica los cálculos de distancia de los objetos a los centroides utilizando la norma de distancia Euclidiana.

El óptimo agrupamiento de los datos *X* está definido como pares de (U,V) que minimiza localmente a J_m teniendo como resultado las Expresiones 3 y 4.

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\left\|x_i - v_j\right\|_2^2}{\left\|x_i - v_k\right\|_2^2}\right)^{\frac{1}{(m-1)}}}; \qquad \begin{matrix} 1 \leq i \leq n; \\ 1 \leq j \leq c \end{matrix} \tag{3}$$

$$v_j = \frac{\sum_{i=1}^{n} (\mu_{ij})^m x_i}{\sum_{i=1}^{n} (\mu_{ij})^m}; \qquad 1 \leq j \leq c \tag{4}$$

Donde x_i y v_j son vectores que pertenecen al espacio $\mathcal{R}^d$ y se representan en las Expresiones 5 y 6.

$$x_i = (x_1, x_2, \ldots, x_d) \qquad 1 \leq i \leq n \tag{5}$$
$$v_j = (v_1, v_2, \ldots, v_d) \qquad 1 \leq j \leq c \tag{6}$$

Las restricciones del agrupamiento difuso se representan en las Expresiones 7-10.

$$\mu_{ij} \in [0,1] \qquad 1 \leq i \leq n; \; 1 \leq j \leq c \tag{7}$$

$$\sum_{j=1}^{c} \mu_{ij} = 1 \qquad 1 \leq i \leq n; \tag{8}$$

$$\mu_{ij} = 1 \qquad \text{if } x_i = v_j \tag{9}$$

$$0 < \sum_{i=1}^{n} \mu_{ij} < n \qquad 1 < j \leq c \tag{10}$$

En la Expresión 7, se indica el grado de pertenencia de un objeto *i* a un grupo *j* debe estar entre 0 y 1; en la Expresión 8, define que la suma de los grados de pertenencia de un objeto *i* a todos los grupos debe ser igual a 1; la Expresión 9 indica que el valor de pertenencia de un objeto *i* a un grupo *j* será igual a uno si el objeto x_i se encuentra en la misma posición que el centroide v_j; la Expresión 10, indica que la suma de todos los grados de pertenencia de los objetos a un grupo debe ser mayor a cero y menor a *n*, es decir no puede haber grupos en los que la suma de los grados de membresía de los objetos al centroide sea cero, y tampoco únicamente un grupo. En la Figura 5, se muestra un ejemplo de la partición realizada con Fuzzy *C*-Means.

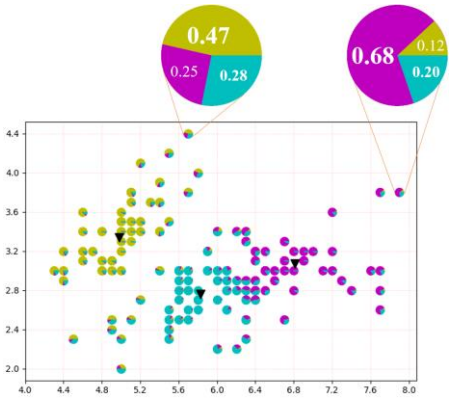


Figura 5. Agrupamiento con Fuzzy *C*-Means.

2.2.1 Fase de inicialización

La fase de inicialización del algoritmo Fuzzy C-Means consiste en asignar los valores a los parámetros para realizar el agrupamiento. Esta etapa es crítica, ya que influye directamente en la calidad de la solución final y en la velocidad de convergencia del algoritmo. En esta fase se definen los siguientes elementos:

- El número de grupos *c*, donde $c \geq 2$, el cual determina la cantidad de particiones en las que se dividirá el conjunto de datos.
- El parámetro difuso *m*, donde $m > 1$, que controla el grado de difusidad en la asignación de pertenencia de los objetos a los grupos.
- El umbral de convergencia ε, el cual establece el criterio de paro del algoritmo.
- La matriz de pertenencia inicial $U^{(0)} = [u_{ij}]$, donde cada elemento representa el grado de pertenencia del objeto x_i al grupo j.

La matriz de pertenencia inicial se genera comúnmente de manera aleatoria, cumpliendo las restricciones del agrupamiento difuso establecidas en las Expresiones (8–11). Es decir, para cada objeto x_i, la suma de sus grados de pertenencia a todos los grupos debe ser igual a uno. Una alternativa a la inicialización aleatoria es la selección de centroides iniciales $V^{(0)}$, a partir de los cuales se puede derivar la matriz de pertenencia. No obstante, en el enfoque tradicional de Fuzzy C-Means, la inicialización se realiza sobre la matriz *U*, debido a su relación directa con la función objetivo.

2.2.2 Fase de clasificación

En la fase de clasificación, el algoritmo Fuzzy C-Means actualiza los grados de pertenencia de cada objeto a los diferentes grupos, con base en la distancia entre los objetos y los centroides actuales. Esta etapa es fundamental, ya que define la estructura difusa de los grupos.

El cálculo de la matriz de pertenencia $U = [u_{ij}]$ se realiza utilizando la Expresión (4), la cual depende de la distancia entre cada objeto x_i y cada centroide v_j. De esta manera, un objeto tendrá mayor grado de pertenencia al grupo cuyo centroide esté más cerca. En la mayoría de los casos, se utiliza la norma Euclidiana como medida de distancia, aunque pueden emplearse otras métricas dependiendo del problema.

Formalmente, esta fase busca minimizar la función objetivo establecida en la Expresión (3) mediante la actualización de los grados de pertenencia, manteniendo las restricciones del modelo difuso. Este proceso realiza un agrupamiento suave, en contraste con otros métodos de partición como K-Means.

2.2.3 Fase de recálculo de centroides

En la fase de recálculo de centroides se obtienen nuevos valores para el conjunto de centroides $V = \{v_1, v_2, \dots, v_c\}$, con el objetivo de reflejar la nueva distribución de los datos. El cálculo de cada centroide v_j se realiza utilizando la Expresión (5), la cual corresponde a una media ponderada de los objetos del conjunto de datos. En dicha ponderación, los pesos están determinados por los grados de pertenencia elevados al parámetro difuso *m*.

Este procedimiento puede interpretarse como una generalización del cálculo de centroides en K-Means, donde en lugar de considerar asignaciones binarias, se utilizan grados de pertenencia difusos. De esta forma, cada objeto contribuye a la actualización de todos los centroides, en diferente proporción.

La actualización de los centroides tiene como objetivo reducir el valor de la función objetivo, ajustando la posición de los grupos en el espacio de características. Este proceso, en conjunto con la fase de clasificación, se realiza en cada iteración del algoritmo Fuzzy C-Means. Es importante mencionar que la precisión en el cálculo de los centroides impacta directamente en la calidad de la agrupación, ya que define la representación de cada grupo en el espacio de datos.

2.2.4 Fase de convergencia

En la fase de convergencia se determina el criterio de terminación del algoritmo Fuzzy C-Means. En esta fase se pueden incluir una o dos condiciones de paro. Existen tres condiciones de paro:

- Alcanzar un número máximo de iteraciones.
- Que la diferencia en la matriz de pertenencia en dos iteraciones sea menor a un umbra.
- Que la diferencia de los valores de los centroides en dos iteraciones sea menor a un umbral.

Debido a que el algoritmo es iterativo, es necesario establecer condiciones bajo las cuales se considere que se ha alcanzado una solución estable. Las Expresiones (11) y (12) muestran dos indicadores de convergencia (Bezdek. J., 1973) cuyos valores se utilizan para detener el algoritmo cuando se cumple una condición de paro:

$$\Delta U^{(t)} = max\left(abs\left(u_{ij}^{(t-1)} - u_{ij}^{(t)}\right)\right) \tag{11}$$

$$\Delta V^{(t)} = max\left(abs\left(v_{ij}^{(t-1)} - v_{ij}^{(t)}\right)\right) \tag{12}$$

En términos generales, el algoritmo converge cuando el cambio entre iteraciones consecutivas es menor a un umbral definido, lo que indica que se ha alcanzado un mínimo local de la función objetivo. Sin embargo, uno de los principales problemas en esta fase es que el algoritmo puede continuar iterando aun cuando la reducción en el valor de la función objetivo es mínima. Esto genera un alto costo computacional, especialmente en conjuntos de datos de gran tamaño. Adicionalmente, la selección del valor de ε no siempre está claramente justificada en la literatura, lo que puede afectar el equilibrio entre calidad de solución y eficiencia computacional.

Por lo anterior, la fase de convergencia representa un punto clave para la optimización del algoritmo Fuzzy *C*-Means, ya que un criterio de paro adecuado puede reducir significativamente el número de iteraciones sin comprometer la calidad de la solución.

Capítulo 3

3. Estado del arte

En este capítulo se presentan de forma sintetizada, investigaciones relevantes sobre mejoras al algoritmo Fuzzy C-Means. Las investigaciones revisadas fueron clasificadas en tres secciones: Fase de inicialización, Fase de clasificación y Fase de convergencia. Así mismo, se reportan los antecedentes de la presente investigación.

3.1 Antecedentes

En CENIDET se han desarrollado diversas tesis que anteceden a la presente investigación con respecto al algoritmo Fuzzy *C*-Means, a continuación, se muestra algunas de ellas.

En (Roblero, S., 2023) se describe el desarrollo de una mejora del algoritmo Fuzzy *C*-Means. El objetivo de la tesis fue reducir el tiempo de procesamiento del algoritmo Fuzzy *C* -Means al resolver conjuntos de datos grandes. Se integró el uso de dos variantes del algoritmo K-Means; la primera K++ (Arthur, D. y Vassilvitskii, S., 2007) se usa para seleccionar centroides iniciales optimizados mediante una probabilidad de distancia, calculada de los objetos más alejados entre sí, posteriormente se usa O-K-Means (Pérez, J. et al., 2018) para agrupar el conjunto de datos mediante un método de convergencia eficiente de K-Means; los centroides finales se vuelven los centroides iniciales para Fuzzy *C*-Means, los centroides son procesados tras un método de transformación para convertirse en la matriz de pertenencia inicial. Este algoritmo se denominó Hybrid OK-Means Fuzzy C-Means (HOFCM) y se validó de manera experimental con conjuntos de datos reales y sintéticos. Se contrastaron los resultados con los algoritmos FCM-KMeans, FCM++ y NFCM, y se observó que HOFCM fue más rápido 1.51, 2.87 y 3.01 veces, respectivamente.

En (Vela, V., 2022) se describe la implementación de los conjuntos difusos titubeantes a un algoritmo de agrupamiento particional. Debido a la incertidumbre de los grados de pertenencia, existen distintos valores posibles que se pueden asignar a un objeto con respecto a muchos grupos. El algoritmo propuesto en esta investigación se denominó Hesitant Fuzzy C-Means, además, se le realizaron tres mejoras: programación paralela usando la técnica OpenMP, asignación de número de grupos mediante el índice Calinski-Harabasz y la generación inicial de la matriz difusa titubeante mediante funciones de pertenencia. El algoritmo propuesto se aplicó a la segmentación de imágenes y al reconocimiento de patrones, y se validó de manera experimental con conjuntos de datos reales y distintas métricas de evaluación. Los resultaron se contrastaron con los algoritmos AKFCMS, FRFCM, DSFCM_N, AFCF, RSSFCA, FCM_SICM, Fuzzy *C*-Means, Kmedoid, HPSOFCM, KMM y FCM-ELPSO, y se observó que las mejoras propuestas tienen mejor desempeño y rendimiento en aplicaciones y comparaciones realizadas.

En (Rey, C., 2023) se describe la implementación de una mejora del algoritmo Fuzzy C-Means en programación paralela. El problema de agrupamiento del algoritmo Fuzzy *C*-Means es de tipo *NP-Hard*, y tiene una complejidad computacional de tipo exponencial, ocasionando que, al resolver conjuntos de datos grandes, el tiempo de solución sea grande. Esta tesis se enfoca en resolver conjuntos de datos grandes, usando todos los procesadores de un equipo de cómputo y con ello reducir los tiempos de solución. El algoritmo propuesto en esta investigación utiliza la técnica OpenMP, se denominó Parallel Optimization Fuzzy C-Means (POFCM) y se validó de manera experimental. Se contrastaron resultados con el algoritmo Fuzzy *C*-Means y HOFCM y se observó que POFCM es más rápido en los tiempos de solución de conjuntos de datos reales.

3.2 Trabajos relacionados

3.2.1 Fase de inicialización

En la fase de inicialización de Fuzzy *C*-Means, se seleccionan los centroides iniciales para formar los c grupos, se define el valor del factor difuso m y se define el valor del umbral de paro ε, el cual mientras más bajo sea su valor, tenderá a obtener mejores resultados, a costa de que el algoritmo realice más iteraciones.

En (Stetco, A. et al., 2015), se obtuvo ventaja de la similitud entre el algoritmo K-Means y el Fuzzy *C*-Means. En particular, para la selección de centroides iniciales, se adaptó una mejora al algoritmo K-Means, nombrada K-Means++ (Arthur, D. y Vassilvitskii, S., 2007). El algoritmo propuesto en esta investigación se nombró FCM++. Posteriormente, a esta generación de centroides, se continúa con el algoritmo Fuzzy *C*-Means estándar.

En (Siringoringo, R. y Jamaluddin, J., 2018) se describe una mejora al algoritmo Fuzzy *C*-Means. Los autores implementaron un algoritmo heurístico denominado Optimización por enjambre de partículas, el cual se enfoca en la selección de los centroides iniciales de Fuzzy *C*-Means. Su rendimiento lo evalúan con 3 métricas denominadas Índice Rand, Medida-F y Función objetivo, con ello los autores determinaron que la propuesta realiza una convergencia en menor tiempo que el algoritmo convencional. En sus resultados destacan una mejora en el valor de la función objetivo de 0.5% con respecto a Fuzzy *C*-Means estándar.

En (Liu, X. et al., 2019) se describe una mejora al algoritmo Fuzzy *C*-Means. Con el fin de reducir la cantidad de iteraciones, obtener una convergencia más rápida y evitar las soluciones óptimas locales. Proponen usar el algoritmo de agrupación por picos de densidad (DPC) para la búsqueda de los centroides iniciales y los grupos con los que se solucionará el conjunto de datos. Entre los resultados que reporta, describe una precisión de 84.99% con un conjunto de datos denominado *Jain* y un valor en el índice ajustado rand de 48.48%, siendo estos valores mejores que otros algoritmos implementados.

En (Wu, Z. et al., 2019) se utiliza el algoritmo K-Means para generar los centroides finales, los cuales, a través de un proceso de transformación denominado *One-Hot encoding*, generan la matriz de membresía que requiere el algoritmo Fuzzy *C*-Means. Para llegar a dicha codificación, esta investigación propone seleccionar los atributos por medio de ponderación de entropía y transformación *Box-Cox*. Esta publicación destaca resultados en el tiempo de solución, obteniendo una ganancia en el tiempo de 5.19% en el mejor de los casos y una mejora de 5.56% en coeficiente de silueta con respecto a Fuzzy *C*-Means estándar.

En (Liu, Q. et al., 2020), toma la estrategia del algoritmo FCM++. En esta investigación se propone el algoritmo denominado NFCM. La diferencia con el FCM++, radica en la consideración de la información que provee la matriz de membresía para la

selección de los siguientes centroides. En (Liu, Q. et al., 2021), extienden la investigación (Liu, Q. et al., 2020). El aporte de esta publicación radica en la modificación de los valores del factor difuso del algoritmo Fuzzy *C*-Means, comparando sus resultados con el algoritmo FCM++ obteniendo un menor valor en la función objetivo y menos iteraciones.

En (Yi, C. et al., 2021) se realiza una mejora a Fuzzy *C*-Means basándose del método t-SNE, el cual realiza una reducción de dimensionalidad de un conjunto de datos y la selección de un centro inicial para una agrupación más precisa que el método convencional, la finalidad es evitar una agrupación con centroides iniciales seleccionados de manera aleatoria, ya que puede converger en un óptimo local. Este estudio destaca resultados favorables en la métrica de precisión del algoritmo obteniendo un valor de 0.96 en el mejor de los casos.

En (Zhang, X. et al., 2021) se describe una mejora a dos fases del algoritmo. El primero es a la manera en que se inicializan los centroides, el cual es a partir de la detección de picos, con ello se obtiene centroides distribuidos de una manera uniforme entre los rangos de datos. La segunda mejora es en la función objetivo, que combina información no local y una prioridad de bajo rango, su aplicación es principalmente en la segmentación de imágenes. Uno de los resultados que destaca es el tiempo de solución obteniendo una reducción de 32.98%.

En (Mohammad, S. et al., 2021) se describe una mejora al algoritmo Fuzzy *C*-Means. Los autores implementan un algoritmo heurístico denominado colonia de abejas, el cual realiza diferentes soluciones posibles a partir de objetos seleccionados entre un mínimo y máximo de cada atributo de datos, se evalúa una función de aptitud a cada individuo, a través de la función objetivo de Fuzzy *C*-Means. Entre sus principales resultados destacan el índice ajustado rand (ARI) con el cual obtuvieron un valor de 0.83, siendo mejor que otras variantes de Fuzzy *C*-Means implementadas.

En (Mújica, D. y Rubio, J., 2021) se propone una mejora al algoritmo Fuzzy *C*-Means intuitivo. Los autores proponen dos etapas de mejora en la fase de inicialización. Esto consta en realizar una transformación a los colores de una imagen a colores cromáticos y se busca en ellos el color dominante para ser usados como los centroides iniciales. Este método fue usado para la segmentación de imágenes y reporta reducción en el tiempo de procesamiento,

además de buenos resultados en métricas de evaluación, representando 1.67% de error menor que otros algoritmos.

En (Pérez, J. et al., 2022) se describe la implementación de una variante híbrida del algoritmo K-Means denominada O-K-Means (Pérez, J. et al., 2018). El aporte consiste en generar una cantidad de 10 soluciones de un conjunto de datos *X* usando K++ para seleccionar centroides optimizados y resolver el conjunto de datos con O-K-Means, posterior a ello se seleccionan los centroides con los que se obtuvo menor valor en la función objetivo, y se usan como base para optimizar los valores de la matriz de membresía y resolver *X* con el algoritmo Fuzzy *C*-Means. Esta publicación destaca resultados favorables en el tiempo y calidad de solución con una reducción de hasta 93.94% en el tiempo y una mejora en la calidad con 68.31% con respecto a Fuzzy *C*-Means estándar.

En (Chen, Y. et al., 2022) se realiza una mejora a Fuzzy *C*-Means variando el parámetro difuso, que es un parámetro de inicialización del algoritmo y afecta directamente al rendimiento del mismo. Se basan en el hecho de que Fuzzy *C*-Means es complejo cuando este valor es grande, o cuando este valor es más cercano a uno tiende a terminar en óptimos locales.

3.2.2 Fase de clasificación

En la fase de clasificación, el algoritmo realiza los cálculos de distancias de los objetos a los centroides, existen diferentes métricas para calcular distancia como es la norma Euclidiana, la norma de la Diagonal, la norma Mahalanobis, entre otros. Con base a la distancia se calcula la matriz de membresía de los objetos a los centroides.

En (Wang, X. *et al*., 2004) se describe una mejora del algoritmo Fuzzy *C*-Means, los autores toman en cuenta las medidas de las características de los conjuntos de datos, se usa la norma Euclidiana e incluye en la función objetivo del algoritmo. Las pruebas que realiza en experimentación obtienen el siguiente aporte: definen la cantidad de grupos necesario para agrupar un conjunto de datos, dependiendo de los pesos de los atributos. En sus resultados destacan también el tiempo de solución, logrando una reducción de hasta 88.53% con respecto al método convencional.

Un trabajo similar es descrito en (Lan, R. y Fan, J., 2009) esta publicación abarca los valores de peso de los atributos de los conjuntos de datos, al igual que los pesos de los grupos, la medición de los pesos se realiza durante el proceso de la agrupación y de estas dos aportaciones se probaron con instancias sintéticas, reportando una mejora en la agrupación, sin importar como se inicialicen los centroides.

En (Mújica, D. et al., 2013) se describen dos mejoras al algoritmo Fuzzy *C*-Means. Los autores describen que se puede obtener información espacial entre los pixeles por medio de los estimadores (RML) y L. El primero presenta la información espacial por medio de una evaluación de los pixeles que se encuentran dentro de un rango definido, esto está basado para la segmentación de imágenes usando colores RGB. El segundo aplica este método, pero su evaluación está centrada en los colores cromáticos y la relación que tienen con la escala RGB.

En (Lee, G. y Gao, X., 2021) se propone un algoritmo con enfoque híbrido, combinando al algoritmo Fuzzy *C*-Means, el algoritmo Genético y una red de retro-propagación para predecir el tiempo del ciclo de trabajo en la manufactura de semiconductores. En (Verma, H. et al., 2021; Hanane, B. y Abdeljabbar, C., 2015; Weiling, C. et al., 2007) se describen aportes para la segmentación de imágenes; la primera publicación describe un algoritmo híbrido entre Fuzzy *C*-Means y *Particle Swarm Optimization* (PSO), los cuales combinan las características de los algoritmos principalmente para la optimización de los centroides para el algoritmo Fuzzy *C*-Means, el uso de PSO genera una mejor comparación de los datos de imágenes cerebrales. Por otro lado, algunos autores describen la conveniencia de introducir una variable que controle la compensación de la agrupación que realizan dos variantes del algoritmo Fuzzy *C*-Means, ya que se puede mejorar los tiempos de solución y que error de la segmentación sea menor al que realiza el algoritmo estándar, usando el valor de x_i como los promedios de escalas de grises.

En (Khang, T. et al., 2021) se describe una mejora al algoritmo Fuzzy *C*-Means. Se describe la integración de una matriz auxiliar, la cual ajusta los grados de pertenencia de los objetos. Sin embargo, cambian el valor del parámetro difuso a través de las iteraciones, con ello estos ajustes permiten que los objetos tengan una mayor pertenencia a ciertos grupos durante la fase de clasificación. Los autores reportan ejemplos en los que identifican cambios

en los grados de pertenencia, identificando un incremento en comparación con el método convencional.

En (Antoine, V. et al., 2022) se describe una mejora al algoritmo Fuzzy *C*-Means. Los autores describen la importancia de mejorar el algoritmo posibilista, por ser ideal en la agrupación de datos con presencia de ruido. Añaden al algoritmo información espacial, para combinar patrones etiquetados en el marco posibilista. Además, utilizan una norma de distancia denominada Mahalanobis.

3.2.3 Fase de convergencia

En la etapa de convergencia del algoritmo, se pueden evaluar una o varias condiciones para llegar a la convergencia. El algoritmo puede converger cuando la cantidad de iteraciones que realiza es igual a la que define un usuario, otra forma de detener el algoritmo, es definiendo un valor a la variable épsilon (Ɛ), el cual es comparado con el valor absoluto de la máxima diferencia entre los grados de pertenencia de todos los objetos hacia los centroides en dos iteraciones consecutivas. Otra opción para evaluar el criterio de parada es comparar el valor de la variable épsilon (Ɛ) con el valor absoluto de la máxima diferencia en los valores de los centroides en dos iteraciones consecutivas. Si la diferencia es menor al de Ɛ el algoritmo converge.

Existen diferentes mejoras al algoritmo Fuzzy *C*-Means, y que definen el valor Ɛ = 0.00001 para sus experimentos como se mencionan en (Dunn, J., 1974; Liu, Q. et al., 2021; Lan, R. y Fan, J., 2009; Lin, Z. et al., 2009; Mahdi, H. et al., 2019), otras investigaciones como las de (Yuan, B. et al., 2002; Parker, J. y Hall, L., 2014) usan Ɛ = 0.001. Sin embargo, de las publicaciones que se encontraron no definen una razón para usar esos valores en la experimentación que realizan.

En (Cebeci, Z., 2018) se describe una mejora la cual reduce la cantidad de iteraciones y el esfuerzo computacional. La propuesta se basa en la búsqueda de la característica con mayor variabilidad en un conjunto de datos, puede ser usando medidas de variabilidad como la desviación estándar o el coeficiente de variación, después se crea una distribución de frecuencias con cierto número de grupos. Por último, la matriz de pertenencia se calcula dividiendo las desviaciones absolutas entre las sumas de filas para cada objeto. Los autores

reportan una reducción en el tiempo de hasta 10.52% con respecto al algoritmo Fuzzy *C*-Means.

En (Hall, L. y Goldgof, D., 2011) se demuestra que la ponderación de distintos conjuntos de datos no causa problemas en la convergencia de Fuzzy *C*-Means. Se describe el uso de un valor en la función objetivo, que son los pesos de los grupos, la cual se obtiene sumando los valores de los grados de pertenencia de cada objeto a un cierto grupo. Al usar este valor se pretende realizar una convergencia óptima cuando este valor es entero. Los autores no reportan un entorno de pruebas.

3.3 Métricas de evaluación utilizadas con algoritmo Fuzzy *C*-Means

En la literatura especializada se observó la aplicación de diferentes métricas de evaluación usadas con el algoritmo Fuzzy *C*-Means. Algunas métricas evalúan la estructura interna de los datos agrupados, mientras que otras evalúan la calidad del agrupamiento. A continuación, se describen algunas de ellas.

3.3.1 Métricas internas

Las métricas internas, evalúan la estructura de los grupos generados por Fuzzy *C*-Means. Uno de ellos es el Coeficiente de Partición (PC), que evalúa el grado de solapamiento entre los grupos generados por Fuzzy *C*-Means. Se calcula a partir de los valores de pertenencia de los objetos a los grupos. La expresión 13 muestra el PC:

$$PC = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^{2} \tag{13}$$

donde u_{ij} representan el grado de pertenencia del objeto i al grupo j. Esta métrica maneja valores de 0-1, mientras más cercano a uno indica grupos bien definidos, un valor más cercano a cero indica mayor solapamiento entre grupos

La Entropía de Partición (PE) mide la incertidumbre presente en la matriz de pertenencia. La expresión 14 muestra el PE:s

$$PE = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}\log(u_{ij}) \tag{14}$$

donde u_{ij} representan el grado de pertenencia del objeto *i* al grupo *j*. Esta métrica maneja valores de 0-1, mientras más cercano a uno hay mayor incertidumbre, valores más cercanos a cero índica una agrupación bien definida.

La métrica de *Silhouette Coefficient* (SC) evalúa la compacidad interna y la separación entre grupos. La expresión 15 muestra el SC:

$$SC = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{15}$$

donde *a*(*i*), representa la distancia promedio del objeto a los elementos de su grupo, *b*(*i*), representa distancia promedio al grupo vecino más cercano. Esta métrica maneja valores de 0-1, mientras más cercano a uno muestra una mejor compacidad y separación entre grupos, un valor más cercano a cero indica grupos cercanos entre sí, y poca compacidad.

3.3.2 Métricas externas

Las métricas externas, evalúan la similitud entre los grupos formados y una clasificación previa de los datos. Uno de ellos es la métrica de *Accuracy*, que mide de la proporción de objetos correctamente clasificados respecto a las etiquetas reales. La expresión 16 muestra el *Accuracy*:

$$Accuracy = \frac{TP + TN}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{16}$$

donde *TP*, son los elementos que realmente pertenecen a una clase y fueron correctamente identificados; *TN*, son los elementos que no pertenecen a una clase y fueron correctamente descartados; *FN*, son los elementos que fueron asignados incorrectamente a una clase; *FN*, son los elementos que pertenecían a una clase, pero no fueron detectados.

La métrica *Rand Index* (RI) similar al *Accuracy* pero esta métrica evalúa la similitud entre dos particiones considerando pares de objetos. La expresión 17 muestra el RI:

$$RI = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

donde *TP*, son dos objetos que pertenecen a la misma clase real y fueron agrupados juntos; *TN*, representa dos objetos que pertenecen a clases distintas y además fueron colocados en grupos diferentes; *FP*, son dos objetos que pertenecen a clases distintas, pero

el algoritmo los coloca en el mismo grupo; y *FN*, son dos objetos que pertenecen a la misma clase real, pero el algoritmo los coloca en grupos distintos.

La métrica *Adjusted Rand Index* (ARI) compara la similitud entre dos particiones considerando el acuerdo esperado por azar. La expresión 18, muestra el ARI:

$$ARI = \frac{RI + E(RI)}{\max(RI) - E(RI)} \tag{18}$$

donde *RI*, es *Rand Index*; *E*(*RI*), es el valor promedio esperado si las etiquetas se hubiesen asignado de manera aleatoria.

La métrica *Normalized Mutual Information* (NMI), mide la cantidad de información compartida entre la agrupación obtenida y las etiquetas reales. La expresión 19, muestra el NMI:

$$NMI = \frac{2I(X,Y)}{H(X) + H(Y)} \tag{19}$$

donde *I(X,Y)*, es la información mutua; *H(X)*, *H(Y)*, son las entropías de cada partición. Esta métrica usa valores entre 0-1, un valor cercano cero indica una baja similitud entre dos particiones, y un valor cercano a uno, indica una alta similitud entre dos particiones.

La métrica *Precision* (P), indica la proporción de objetos asignados correctamente dentro de una clase predicha. La expresión 20, muestra el P:

$$P = \frac{TP}{TP + FP} \tag{20}$$

Esta métrica usa valores entre 0-1, un valor cercano a cero, indica una mala clasificación en los datos, y un valor cercano a uno indica que los datos fueron clasificados correctamente.

La métrica *Recall* (R), mide la capacidad para agrupar todos los objetos que pertenecen a una misma clase. La expresión 21, muestra el R:

$$R = \frac{TP}{TP + FN} \tag{21}$$

Esta métrica usa valores entre 0-1, un valor cercano a cero indica que los datos que pertenecen a una misma clase están siendo colocados en distintos grupos, y un valor cercano a uno indica que fueron agrupados en un mismo grupo y que pertenecen a la misma clase.

3.4 Conclusiones del estado del arte

En la construcción del estado del arte, se revisaron diferentes documentos científicos sobre mejoras al algoritmo Fuzzy C-Means. Entre ellos, artículos publicados en revistas y congresos, capítulos de libros, *surveys*, entre otros. Se observó un amplio estudio del algoritmo Fuzzy C-Means en su fase de inicialización, mientras que la fase de convergencia es la menor abarcada, lo que representa una oportunidad para contribuir a su investigación.

Se revisaron múltiples investigaciones, que abarcan la fase de inicialización, las cuales proponen diversos métodos que optimicen los parámetros iniciales, como la matriz de membresía, centroides iniciales, los valores del factor difuso *m*, y evitar la dependencia a la aleatoriedad. Entre esas técnicas se encuentra el uso de algoritmos como K++ y O-K-Means, métodos heurísticos como colonia de abejas y PSO, métodos basados en densidad como DPC y técnicas de estadística como selección basada en la entropía o picos de densidad. Estas implementaciones, demuestran impactos positivos en la comunidad científica por la reducción de tiempos de procesamiento y mejoras en la calidad de solución. Mientras que la fase de convergencia ha sido la menos explorada, lo que representa una oportunidad valiosa para contribuciones novedosas.

Se observó que la fase de convergencia, es la menos explorada y requiere mayor investigación. Las principales propuestas, se enfocan en reducir el número de iteraciones para alcanzar la convergencia y en la optimización del criterio de paro épsilon (ε). Aunque varios estudios establecen diversos valores de ε no se encontró una justificación de su elección, siendo una oportunidad para investigación.

Finalmente, se observó que el algoritmo Fuzzy C-Means es relevante para su investigación. Mayormente en la fase de convergencia, ya que es la menos explorada para sustentar criterios de parada robustos. Esto respalda la importancia de enfocar el estudio a la fase de convergencia, para una mejora al algoritmo Fuzzy C-Means más eficiente y estable. La Tabla 2, muestra una tabla comparativa de los artículos revisados.

Tabla 2. Comparativa entre artículos de investigación.

Cita	Artículo	Etapa de mejora	Algoritmos de comparación	Complejidad	Reducción de Iteraciones	Mejora en calidad de solución	Métricas de evaluación
(Stetco, A. et al., 2015)	Fuzzy C-means++: Fuzzy C-means with effective seeding initialization	Inicialización	FCM	No especifica	84.5%	XI = No especifica FO = No especifica	Xie-Beni Función objetivo
(Liu, Q. et al., 2020)	A Novel Initialization Algorithm for Fuzzy C-means Problem	Inicialización	k-means++ FCM++	$O(k^2 \ln k)$	No especifica	No especifica	No especifica
(Liu, Q. et., 2021)	Approximation algorithms for fuzzy C-means problem based on seeding method	Inicialización	k-means++ FCM++	$O(k^2 \ln k)$	No especifica	No especifica	No especifica
(Pérez, J. et al., 2022)	Hybrid Fuzzy C-Means Clustering Algorithm oriented to Big Data Realms	Inicialización	FCM FCM++ NFCM	$O(ndc^2t)$	No especifica	68.12%	Función objetivo
(Chen, Y. et al., 2022)	Improved fuzzy c-means clustering by varying the fuzziness parameter	Inicialización	FCM	No especifica	No especifica	No especifica	F* NMI ARI SC
(Yiu, C. et al., 2021)	Improved fuzzy C-means clustering algorithm based on t-SNE for terahertz spectral recognition	Inicialización	FCM (PCA) FCM (LLE) FCM (LPP)	No especifica	No especifica	0.96	*Accuracy*
(Zhang, Q. et al., 2015)	Distributed fuzzy c-means algorithms for big sensor data based on cloud computing	Inicialización	FCM	No especifica	No especifica	No especifica	*Accuracy*
(Mohammad, S. et al., 2021)	Big data fuzzy C-means algorithm based on bee colony optimization using an Apache Hbase	Inicialización	PCM wPCM HOPFCM	No especifica	No especifica	0.92	ARI
(Siringoringo, R y Jamaluddin, J, 2018)	Initializing the Fuzzy C-Means Cluster Center With Particle Swarm Optimization for Sentiment Clustering	Inicialización	FCM	No especifica	No especifica	R = 0.77 P = 0.97 RI = 0.82 FM = 0.86	*Recall* (R) *Precision* (P) *Rand Index* (RI) *F-Measure* (FM)

						FO = 13070.68	Función objetivo
(Mújica, D y Rubio, J, 2021)	Accelerated intuitionistic fuzzy clustering for image segmentation	Inicialización	AFCF FRFCM SFFCM RSSFCA IFCM	No especifica	No especifica	MCR $\leq$ 9.8 Dice $\geq$ 9.21 JSC $\geq$ 9.17 HD $\leq$ 8.9	MCR Dice JSC HD
(Liu, X, et al., 2019)	Improved fuzzy C-means algorithm based on density peak	Inicialización	FCM K-means DBSCAN NMF	$O(2n^2 + npCL)$	No especifica	Acc = 98.43% ARI = 92.69% NMI = 86.56%	*Accuracy* ARI NMI
(Lee, G. y Gao, X, 2021)	A Hybrid Approach Combining Fuzzy c-Means-Based Genetic Algorithm and Machine Learning for Predicting Job Cycle Times for Semiconductor Manufacturing	Clasificación	LR BPN *k*-means BPN *k*-means fuzzy BPN Chen	No especifica	No especifica	MAE = 16.3% MAPE = 4.6% RMSE = 19.6%	MAE MAPE RMSE
(Verma, H. et al., 2021)	A population based hybrid FCM-PSO algorithm for clustering analysis and segmentation of brain image	Clasificación	PSO FCM FCM-FPSO PSO-KIFCM BBO-FCM FCM-ELPSO	No especifica	No especifica	*Precision* = 1 *Recall* = 1 *Accuracy* = 1	*Recall* (R) *Precision* (P) *Accuracy (Acc)*
(Hanane, B y Abdeljabbar, C, 2015)	Fast Robust Fuzzy Clustering Algorithm for Grayscale Image Segmentation	Clasificación	FCM S_FCM FCM_S1 FCM_S2	No especifica	No especifica	Sp = 4.82 Gn = 4.72	Salt and pepper Gaussian noise
(Weling, C. et al., 2007)	Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation	Clasificación	FCM_S1 FCM_S2 EnFCM	$O(qcI_2)$	No especifica	Sp = 95.41 Gn = 99.91 mn = 99.96	Salt and pepper Gaussian noise mixed noise
(Wang, X. et al., 2004)	Improving fuzzy c-means clustering based on feature-weight learning	Clasificación	FCM	$O(cn^2)$	No especifica	PC = 0.9 PE = 0.8 FS = 48,365,212 XE = 0.23	PC PE FS Xie-Beni

(Lan, R y Fan, J, 2009)	A Fuzzy C-means Type Clustering Algorithm on Triangular Fuzzy Numbers	Clasificación	FCN AFNC	No especifica	No especifica	No especifica	Centroides finales
(Mújica, D. et al., 2013)	A fuzzy clustering algorithm with spatial robust estimation constraint for noisy color image segmentation	Clasificación	SNEM IAFHA HTFCM	No especifica	No especifica	PRI = 0.79 VOI = 1.8 GCE = 0.2 BDE = 3.41	PRI VOI GCE BDE
(Antoine, V. et al., 2022)	Possibilistic fuzzy c-means with partial supervision	Clasificación	PFCM	No especifica	No especifica	No especifica	No especifica
(Khang, T. et al., 2021)	A Novel Semi-Supervised Fuzzy C-Means Clustering Algorithm Using Multiple Fuzzification Coefficients	Clasificación	No especifica	No especifica	No especifica	No especifica	Matriz de membresia
(Yuan, B. et al., 2002)	Evolutionary fuzzy c-means clustering algorithm	Convergencia	No especifica	No especifica	No especifica	PC = 0.96 PE = 0.94	PC PE
(Parker, J y Hall, L, 2014)	Accelerating Fuzzy-C Means Using an Estimated Subsample Size	Convergencia	OFCM SPFCM eFFCM rseFCM	$O(nisc)$	No especifica	9.61	MRI
(Cebeci, Z, 2018)	Initialization of Membership Degree Matrix for Fast Convergence of Fuzzy C-Means Clustering	Convergencia	FCM	No especifica	No especifica	No especifica	No especifica
(Hall, L y Goldgof, D, 2011)	Convergence of the Single-Pass and Online Fuzzy C-Means Algorithms	Convergencia	No especifica	No especifica	No especifica	No especifica	No especifica

Capítulo 4

4. Metodología de solución

En este capítulo, se presentan las observaciones del comportamiento del algoritmo Fuzzy C-Means y las heurísticas desarrolladas para reducir la complejidad del algoritmo.

4.1 Introducción

En la metodología de solución, se describen las actividades realizadas para abordar el planteamiento del problema. En las siguientes secciones, de presentan dos métodos propuestos para reducir la complejidad computacional del algoritmo Fuzzy C-Means. El primer método, se enfoca en la fase de convergencia, la cual determina el momento en el que el algoritmo se detiene y tiende a generar una complejidad muy alta; además es una etapa poco abarcada en la literatura científica. El segundo método, se centra en las fases de inicialización y convergencia. En este caso, se propone una variante hibrida que integra el primer método con una mejora del algoritmo Fuzzy C-Means orientada a optimizar la matriz de pertenencia.

4.2 Enfoque de solución en la fase de convergencia

La fase de convergencia consiste en parar el algoritmo Fuzzy *C*-Means cuando se alcanza el valor del umbral (ε). Este valor es comparado con una condición de paro, generalmente la comparación es con valor absoluto de la diferencia máxima de grados de pertenencia en dos iteraciones consecutivas con el valor del umbral (ε) o por medio de la diferencia máxima que tienen los centroides en dos iteraciones consecutivas (Bezdek, J. et al., 1984). En las expresiones 11 y 12 se muestran estos indicadores de paro.

Se han identificado tres condiciones de paro: número máximo de iteraciones (T); que el valor del indicador ΔU sea menor a un umbral ε; que el valor del indicador ΔV sea menor a un umbral ε y que el valor del indicador $\Delta' J_m$ sea menor a un umbral ε. En los casos donde el criterio de convergencia está compuesto por dos condiciones, con que se cumpla una de ellas, el algoritmo para. En la expresión 22, se muestra el indicador de paro de $\Delta' J_m$.

$$\Delta' J_m^{(t)} = \left(J_m^{(t-1)} - J_m^{(t)} \right) \tag{22}$$

4.2.1 Identificación de patrones de solución en la fase de convergencia

Para encontrar patrones de solución, se analizó el comportamiento de la función objetivo de Fuzzy *C*-Means en la fase de convergencia, se realizaron 4 experimentos. Se agruparon cuatro conjuntos de datos sintéticos, con cada uno se usaron 10 centroides iniciales diferentes, generados aleatoriamente. La Tabla 3, muestra los valores de la función objetivo y la máxima diferencia que tienen la matriz de pertenencia a través de las iteraciones obtenidas de la agrupación.

Tabla 3. Resultados de Fuzzy *C*-Means con un conjunto de datos sintético.

t	Experimento 1		Experimento 2		Experimento 3		Experimento 4	
	Función Objetivo	ΔU	Función Objetivo	ΔU	Función Objetivo	ΔU	Función Objetivo	ΔU
1	21.1512	1	26.1974	1	30.9170	1	127.2067	1
2	18.3421	0.4226	19.9959	0.7303	26.5631	0.4849	116.2628	0.5577
-	-	-	-	-	-	-	-	-
20	17.5703	0.0113	15.6248	0.0148	21.2501	0.0304	80.6417	0.0212
21	17.5695	0.0083	15.6239	0.0126	21.2428	0.0231	80.6092	0.0189
22	-	-	15.6232	0.0116	21.2376	0.0186	80.5823	0.0168
23	-	-	15.6226	0.0106	21.2338	0.0161	80.5599	0.0151
24	-	-	15.6221	0.0099	21.2311	0.0139	80.5411	0.0146
25	-	-	-	-	21.2291	0.0120	80.5252	0.0140
26	-	-	-	-	21.2277	0.0103	80.5118	0.0133
27	-	-	-	-	21.2266	0.0089	80.5005	0.0126
28	-	-	-	-	-	-	80.4908	0.0118
29	-	-	-	-	-	-	80.4826	0.0111
30	-	-	-	-	-	-	80.4756	0.0103
31	-	-	-	-	-	-	80.4696	0.0096

Como se observa en la Tabla 3, se muestran los valores de la función objetivo y la diferencia máxima que hay en la matriz de pertenencia en las últimas iteraciones de cada

experimento. El patrón de solución que se observa es el cambio del valor de la función objetivo y la máxima diferencia, estos valores se reducen más en las dos primeras iteraciones que en las últimas, esto demuestra que el costo computacional es demandante incluso cuando estos valores no cambian mucho.

De acuerdo con las observaciones, se presenta una gráfica del Experimento 2 sobre la tendencia que tiene la función objetivo a través de las iteraciones. Se destaca el cambio mínimo que llega a tener la función objetivo entre una iteración y otra. Es conveniente mencionar que para ver estos valores se realizó una normalización de los resultados de la función objetivo en cada iteración. La Figura 6, muestra esta tendencia.

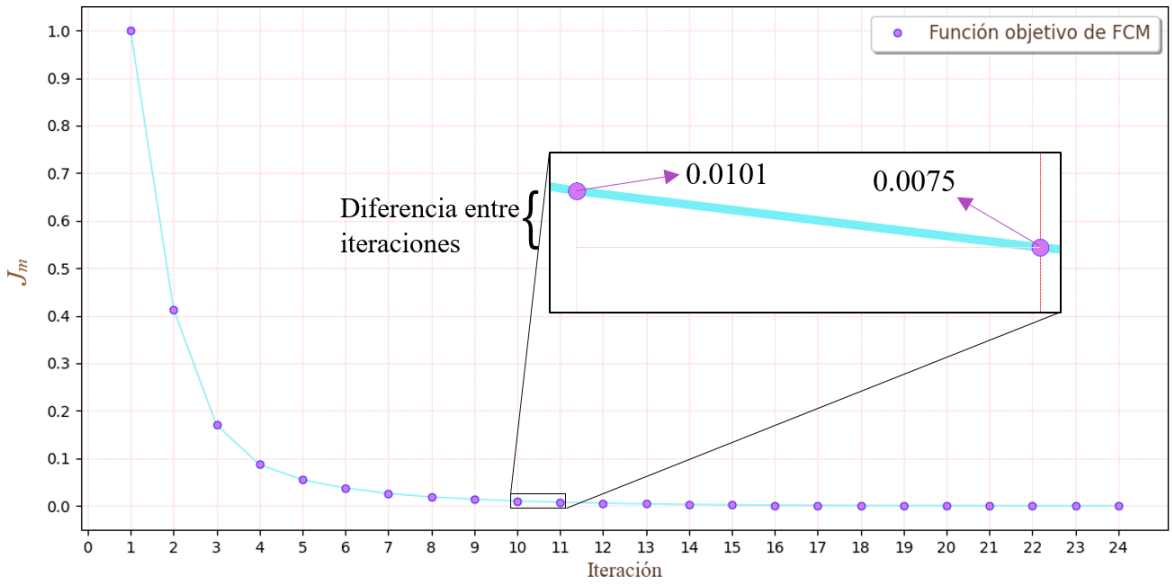


Figura 6. Diferencia de la función objetivo a través de las iteraciones.

Como se muestra en la Figura 6, el valor de la función objetivo en la iteración 10 es de 0.0101 y es mayor a la iteración 11 que es 0.0075. La diferencia que hay en la función objetivo entre la iteración 10 y 11 es de 0.0026. Sin embargo, este cambio en la función objetivo llega a ser poco relevante en comparación con las primeras iteraciones, y el cambio que sufre es similar a las iteraciones futuras, por ello es conveniente evitar que el algoritmo continúe iterando por una ganancia poco significativa, y con ello reducir la complejidad computacional.

4.2.2 Propuesta de heurística P-FCM

Partiendo de las publicaciones que se estudiaron y de los patrones encontrados en la fase de convergencia. Se observó que el indicador ΔU se utiliza con mayor frecuencia, y una de sus limitaciones de este enfoque es que el valor del indicador está desvinculado de los valores de la función objetivo. Esto dificulta la selección de valores de umbral que permitan establecer un balance entre el número de iteraciones y la calidad de solución. En este sentido se propone un nuevo indicador de convergencia basada en los valores de la función objetivo, en dos iteraciones sucesivas, expresada como un porcentaje, ver Expresión (23).

$$\Delta J_m^{(t)} = 100 * \left(\frac{J_m^{(t-1)} - J_m^{(t)}}{J_m^{(t+1)}} \right) \tag{23}$$

Con la finalidad de contrastar los valores de $\Delta J_m^{(t)}$ y $\Delta U^{(t)}$ se resolvieron distintos conjuntos de datos y en general se observó que están altamente correlacionados, en todos los casos la correlación fue mayor a 0.9. El cálculo de correlación se realizó usando las Expresiones 24 y 25 las cuales corresponden a la correlación de Pearson. En particular el conjunto de datos *Abalone* se encontró una correlación de 0.94. Dicho conjunto de datos está conformado por 1,477 registros y siete dimensiones.

$$p(\Delta J_m, \Delta U) = \frac{\sum_{t=2}^{T} [(\Delta J_m^{(t)} - \ddot{\Upsilon})(\Delta U^{(t)} - \ddot{\upsilon})]}{\sqrt{\sum_{t=2}^{T} (\Delta J_m^{(t)} - \ddot{\Upsilon})^2} \sqrt{\sum_{t=2}^{T} (\Delta U^{(t)} - \ddot{\upsilon})}} = 0.94 \tag{24}$$

Donde:

$$\ddot{\Upsilon} = \frac{1}{t} \sum_{t=2}^{T} \Delta J_m^{(t)} \qquad \text{y} \qquad \ddot{\upsilon} = \frac{1}{t} \sum_{t=2}^{T} \Delta U^{(t)} \tag{25}$$

Para validar el uso de este nuevo indicador se buscó la relación de esfuerzo y beneficio entre la cantidad de iteraciones y la calidad de solución. Se utilizó el principio de Pareto como sugieren (X. Liu, et al., 2021), (D. Golovin, 2020) y (M. Kalimuthu, et al., 2023) ya que permite determinar la relación optima entre el esfuerzo y beneficio de resolver un conjunto de datos, una aplicación similar fue con el algoritmo K-Means, publicado por (J. Pérez, et al., 2018). Para la aplicación del principio de Pareto se resolvieron varios conjuntos de datos, ver Tabla 4. Una vez que se obtiene los resultados intermedios de cada iteración, el valor final de la función objetivo y el número de iteraciones se sigue una serie de pasos para obtener una relación óptima entre el número de iteraciones y la calidad de la solución.

Tabla 4. Conjuntos de datos.

Conjunto de datos	Tipo	*n*	*d*	*n*d*
Abalone	Real	4,177	7	29,239
Wine	Real	4,898	11	53,878
Urban	Real	360,177	2	720,354
S3_2	Sintético	36	2	72
S6_2	Sintético	60	2	120
S8_2	Sintético	81	2	162
S10_2	Sintético	100	2	200

El proceso para la búsqueda de una solución óptima del esfuerzo y beneficio, se muestran los resultados de aplicar el principio de Pareto al resolver el conjunto de datos *Abalone* con Fuzzy *C*-Means. Cabe mencionar, que el indicador de paro que se usó fue ΔU, con un valor de $\varepsilon = 0.01$ y un valor de $c = 50$.

Tabla 5. Algunos resultados intermedios de *Abalone*.

Iteración (*t*)	A^t	B^t	C^t	D^t	$\Delta J_m^{(t)}$
1	0.34	--	--	--	--
--	--	--	--	--	--
13	4.45	0.59	96.25	5.81	2.36
14	**4.79**	**0.66**	**96.92**	**5.69**	**2.71**
15	5.13	0.56	97.48	5.72	2.34
--	--	--	--	--	--
17	5.82	0.21	98.03	6.14	0.90
--	--	--	--	--	--
29	9.93	0.03	99.01	9.98	0.15
--	--	--	--	--	--
292	100	0.00	100	100	0.0024

En la Tabla 5, se muestra la relación que existe entre el esfuerzo computacional y la calidad de solución. La columna uno, muestra la cantidad de iteraciones que realizó el algoritmo; la columna dos, muestra el porcentaje acumulado del total de iteraciones y se denota por $A^t = 100 * (t/T)$; la columna tres, muestra la reducción del valor de la función objetivo con respecto a la primera y última iteración, y se denota por $B^t = 100 * (J_m^{(t-1)} - J_m^{(t)})/(J_m^1 - J_m^T)$; la columna cuatro, muestra el acumulado de los porcentajes de reducción del valor de la función objetivo, $C^t = C^{t-1} + B^t$; la columna cinco, muestra el valor de la distancia euclidiana entre los puntos (A^t, C^t) y (0, 100), donde $D^t =$

$\sqrt{(0-A^t)^2+(100-C^t)^2}$; la columna seis, muestra el valor por iteración del indicador de convergencia ΔJ_m.

La Figura 7, muestra el diagrama de Pareto para los puntos (A^t, C^t), ver las columnas dos y cuatro de la Tabla 5. En el eje *A* se muestran porcentajes del total de las iteraciones, en el eje *C* el porcentaje del total de reducciones del valor de la función objetivo. De esta manera el punto G con coordenadas (4.79, 96.92) significa que con el 4.79% de las iteraciones se obtuvo el 96.92% del total de las reducciones del valor de la función objetivo. En caso de parar el algoritmo en este punto solamente se tendría una reducción de la calidad de la solución del 3.08%.

Nótese como el punto t^{14} es el que tiene la menor distancia D^t al punto (0, 100), ubicado en la esquina superior izquierda de la gráfica. De acuerdo con el principio de Pareto dicho punto es el que tiene la relación óptima entre esfuerzo y beneficio. Como se puede observar en el cuarto renglón de la Tabla 5 dichos valores corresponden a los valores de la iteración 14. Para determinar el valor de umbral óptimo se observa en la Tabla 6, sexta columna y tercer renglón, y se observa que el valor de $\Delta J_m^{(t)}$ es de 2.71%. Esto es, cuando la diferencia de la función objetivo en dos iteraciones seriadas sea menor al 2.71% del valor de la función objetivo en la iteración anterior, se para el algoritmo. Es destacable que si el algoritmo para en este punto se evitaría realizar 95.21% de las iteraciones y se tendría el 96.92% del valor acumulado de las reducciones de la función objetivo. La reducción de la calidad sería de solamente un 3.08%.

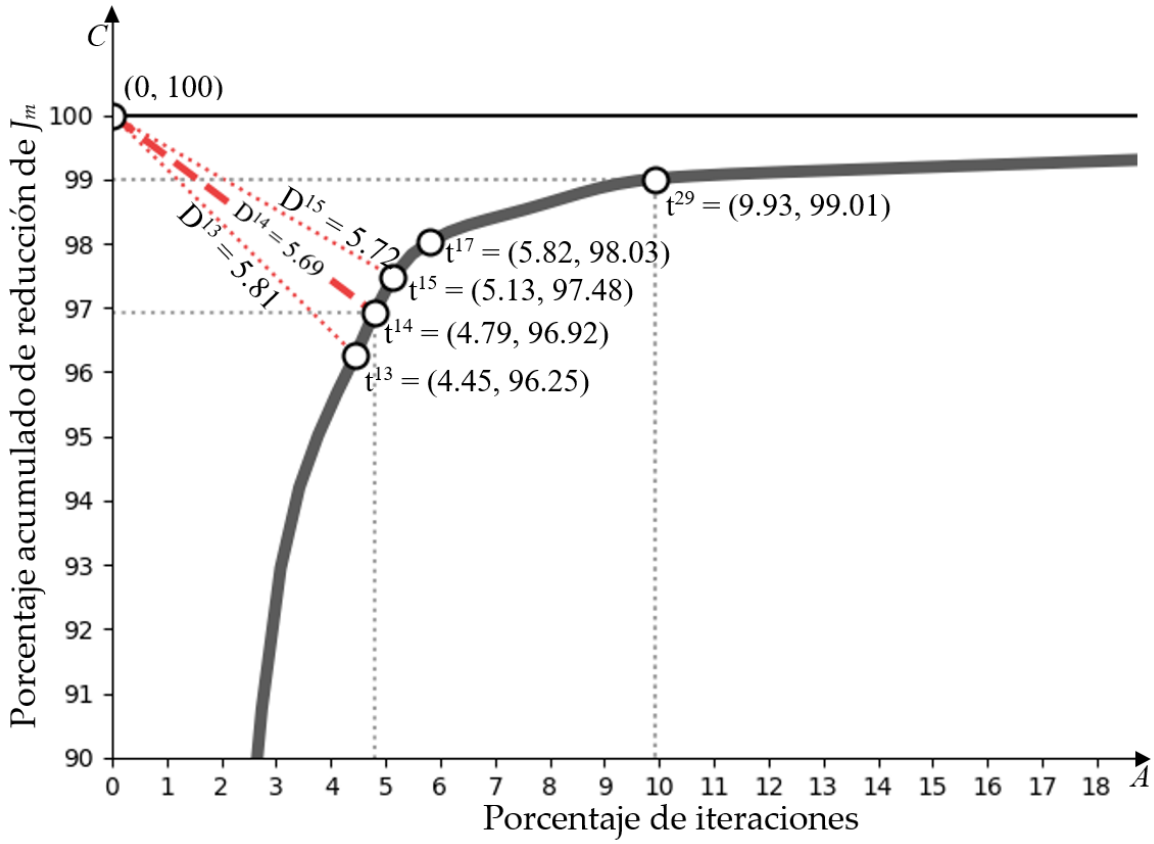


Figura 7. Optimidad de Pareto *Abalone* ε = 0.01.

Nótese como los puntos (t^{13}, t^{14}, t^{15}, t^{17}, t^{29}) tienen reducciones acumuladas de la función objetivo de 96.25, 96.92, 97.48, 98.03 y 99.01 respectivamente, y corresponden a las iteraciones 13, 14, 15, 17 y 29. Como puede observarse en la Figura 7, la relación entre esfuerzo computacional y la calidad de la solución no es lineal. Véase cómo a medida que se desea incrementar la calidad de la solución el número de iteraciones requeridas crece en una proporción mucho mayor.

En la Tabla 6, se muestran algunos resultados intermedios al resolver con Fuzzy *C*-Means el conjunto de datos *Abalone*, con el propósito de contrastar valores de la calidad de la solución y el esfuerzo computacional requerido. En la columna uno, se muestra el número de iteración; en la columna dos, se muestra el porcentaje del total de iteraciones que serían reducidas si el algoritmo para en la iteración; en la columna tres, se muestra el porcentaje acumulado de la función objetivo en la iteración; en la columna cuatro, se muestra el porcentaje de reducción de la calidad en caso de que el algoritmo pare en la iteración; en la columna cinco, se muestra el valor de la función objetivo calculado para la iteración; en la

columna seis, se muestra el valor del indicador ΔJ_m para la iteración; en la columna siete, se muestra el valor del indicador ΔU.

Tabla 6. Resultados de interés al solucionar *Abalone* con Fuzzy *C*-Means.

Iteración	% Reducción Iteraciones	C^t	% Reducción Calidad	J_m	ΔJ_m	ΔU
1	99.65	--	--	28.20	--	0.99
--	--	--	--	--	--	--
14	**95.20**	**96.92**	**3.08**	**5.56**	**2.71**	**0.44**
15	94.86	97.48	2.52	5.43	2.34	0.42
--	--	--	--	--	--	--
17	94.18	98.03	1.96	5.30	0.90	0.22
--	--	--	--	--	--	--
29	90.07	99.01	0.99	5.07	0.15	0.06
--	--	--	--	--	--	--
292	0	100	0	4.84	0.0024	0.0094

En los renglones de la Tabla 6, se muestran valores de algunas iteraciones de interés. Nótese como los valores de la columna tres son aproximadamente del 96, 97, 98, 99 y 100 por ciento y corresponden a las iteraciones 14, 15, 17, 29, y 292 respectivamente. Como se puede observar, pasar del 96 al 97 por ciento de calidad solamente requirió una iteración, sin embargo, pasar del 99 al 100 por ciento requirió de 263 iteraciones. Lo anterior evidencia una posible ineficiencia del algoritmo al conseguir una mejora mínima de la función objetivo a costa de un alto costo computacional.

Por otra parte, de acuerdo con el principio de Pareto la relación óptima entre esfuerzo y beneficio se tiene en la iteración 14. Por lo tanto, se sugiere el valor de 2.71 como umbral de paro usando nuestro indicador propuesto (columna seis), véase como se sugiere el valor de 0.44 como valor de umbral de paro usando el indicador tradicional de Fuzzy *C*-Means (columna siete). Es destacable, que los valores de nuestro indicador son más intuitivos al ser expresados como un porcentaje del valor de la función objetivo, lo cual no ocurre con el indicador tradicional de Fuzzy *C*-Means.

El algoritmo 1 que se muestra a continuación, integra el indicador de paro al algoritmo Fuzzy *C*-Means y se le denominó Pareto Fuzzy *C*-Means (P-FCM). En particular, véase como se integra el nuevo indicador de convergencia en la línea 9.

Algorithm 1 P-FCM:

Input: Dataset X, V, c, m, ε
Output: V, U

Initialization:
  t: = 0;
  $U^{(t)}$: = $\{u_{11}, \ldots, u_{ij}\}$; is randomly generated
Calculate centroids:
  Calculate the centroids using Exp. (5);
Calculate membership matrix:
  Update and calculate the membership matrix using Exp. (4);
Convergence:
  If $\Delta J_m{}^t \leq \varepsilon$: Indicador de convergencia (P-FCM)
    Stop the algorithm;
  Otherwise:
    $U^{(t)}$: = $U^{(t+1)}$ and t: = t + 1;
    Go to **Classification**
End of algorithm

4.3 Enfoque de solución en la fase de inicialización

La fase de inicialización del algoritmo Fuzzy *C*-Means consta en asignar valores a los parámetros de entrada. Los cuales son:

- (m) el factor difuso, este parámetro debe tener un valor mayor a 1.
- (c) la cantidad de grupos, con valor mayor o igual a 2.
- (ε) Épsilon, es el valor del umbral para converger el algoritmo.
- El conjunto de datos, los centroides iniciales o matriz de pertenencia.

En esta sección se describe el desarrollo de una nueva heurística denominada HPFCM. Esta heurística surge de la necesidad de optimizar las fases de inicialización y convergencia para reducir la complejidad computacional, sin afectar la calidad de solución. A continuación, se describe las observaciones en los patrones de solución con el algoritmo Fuzzy *C*-Means y la heurística propuesta para la generación de parámetros iniciales optimizados.

4.3.1 Identificación de patrones de solución en la fase de inicialización

Para encontrar patrones de solución se observó el comportamiento de la función objetivo de Fuzzy *C*-Means en la fase de inicialización. Se realizaron cinco experimentos, en donde se agrupó un conjunto de datos sintético de 400,000 objetos con tres dimensiones cuyos valores están entre 1 - 6, cada agrupación se realizó con 15 diferentes centroides iniciales, generados

de manera aleatoria. Además, se usó un valor de $\varepsilon = 0.01$ como criterio de paro en los cinco experimentos. En la Tabla 7, se muestra estos centroides.

Tabla 7. Generación de centroides iniciales aleatorios para 5 experimentos.

	Experimento 1			**Experimento 2**			**Experimento 3**			**Experimento 4**			**Experimento 5**		
	D^1	D^2	D^3	D^1	D^2	D^3	D^1	D^2	D^3	D^1	D^2	D^3	D^1	D^2	D^3
C^1	6	2	2	3	4	5	6	6	4	3	6	2	4	2	4
C^2	2	2	6	2	2	4	2	3	1	6	3	4	6	1	3
C^3	6	6	2	6	6	6	6	3	5	3	4	6	1	3	6
C^4	4	2	4	2	2	3	2	4	2	1	1	2	1	3	2
C^5	5	3	4	3	4	2	5	6	6	2	5	4	2	4	5
C^6	2	4	2	2	5	1	5	5	2	4	4	3	1	4	3
C^7	3	4	3	3	1	6	3	1	5	2	4	2	4	3	3
C^8	2	3	6	1	1	3	1	6	4	1	5	5	5	5	6
C^9	3	5	1	1	5	5	6	3	3	1	6	4	6	3	3
C^{10}	1	4	6	4	4	2	3	5	2	4	3	1	4	4	2
C^{11}	1	4	4	6	4	4	5	1	6	6	1	3	2	6	1
C^{12}	5	1	1	5	6	5	6	5	1	1	1	4	3	2	6
C^{13}	6	5	5	6	4	4	1	6	3	4	3	4	5	3	1
C^{14}	6	3	6	6	1	5	1	2	2	4	5	4	6	5	6
C^{15}	1	1	1	3	3	1	6	3	4	5	4	6	4	2	4
	Iteraciones = 39			Iteraciones = **27**			Iteraciones = **64**			Iteraciones = 56			Iteraciones = 41		

Como se observa en la Tabla 7, se muestran los centroides iniciales de cada experimento respectivamente. Las columnas D^1, D^2 y D^3 representan los valores de las tres dimensiones de cada experimento; las filas C^1, C^2, …, C^{15} representa un centroide inicial en cada experimento respectivamente; se observa que al usar los centroides del Experimento 2 hace que el algoritmo Fuzzy *C*-Means realice 27 iteraciones (marcado en azul); usar los centroides del Experimento 3 generó 64 iteraciones (marcado en rojo). Esto demuestra que el algoritmo Fuzzy *C*-Means es sensible a la inicialización de sus parámetros iniciales, uno de ellos son los centroides iniciales, lo que provoca un alto costo computacional para conjunto de datos grandes.

La función objetivo de Fuzzy *C*-Means es de minimización y en cada iteración el valor debe ser menor. Considerando los experimentos descritos anteriormente, se analizó el comportamiento de la función objetivo a través de las iteraciones, la Figura 8 muestra el comportamiento de Fuzzy *C*-Means de los cinco Experimentos realizados con diferentes centroides.

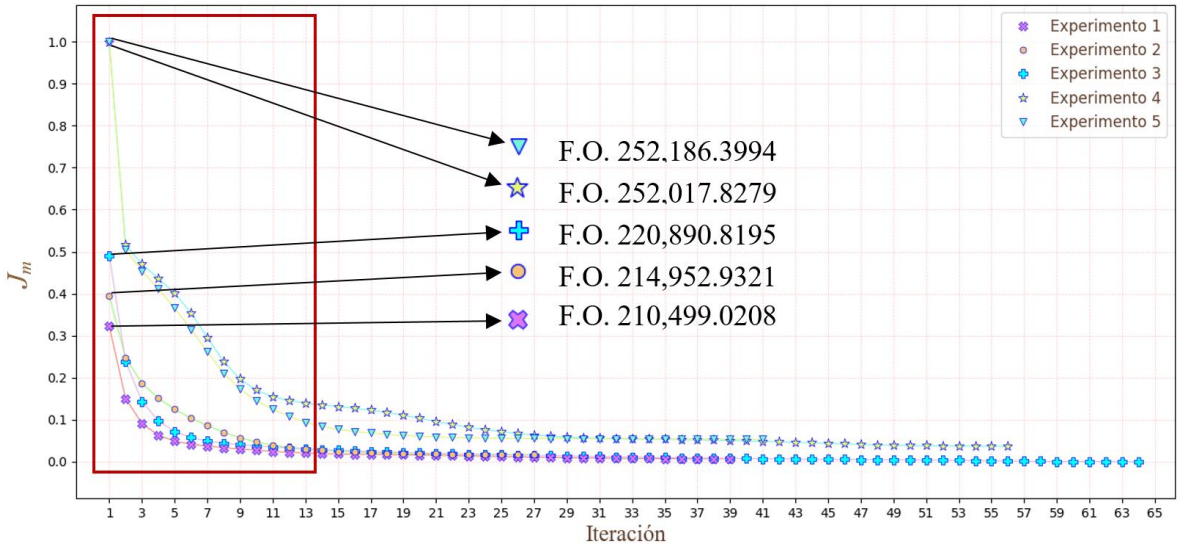


Figura 8. Comparativa de la Función Objetivo de Fuzzy *C*-Means.

Como se observa en la Figura 8, el patrón en el comportamiento de los valores de la función objetivo de Fuzzy *C*-Means, se reduce más en las primeras iteraciones. La diversidad de los valores iniciales se debe a que los centroides iniciales que se usaron fueron generados de manera aleatoria para solucionar el conjunto de datos. Entre los experimentos, el que tuvo la peor inicialización fue el Experimento 5; el experimento que tuvo mejor inicialización fue el Experimento 1 pero terminó en la iteración 39; sin embargo, el experimento 2 tuvo buena inicialización y terminó en la iteración 27. Realizar una selección optimizada de los centroides iniciales puede evitar muchas iteraciones, debido a que los primeros valores de la función objetivo estarán cerca de los valores de convergencia del algoritmo.

4.3.2. Propuesta de heurística HPFCM

Partiendo de las observaciones realizadas para encontrar patrones de solución en la fase de inicialización. Se propone reducir la complejidad computacional de Fuzzy *C*-Means usando centroides optimizados. Esto se logrará por medio del desarrollo de nuevas heurísticas que permitan la selección de buenos centroides a partir de un conjunto de datos. La Figura 9, describe el enfoque de solución de la propuesta.

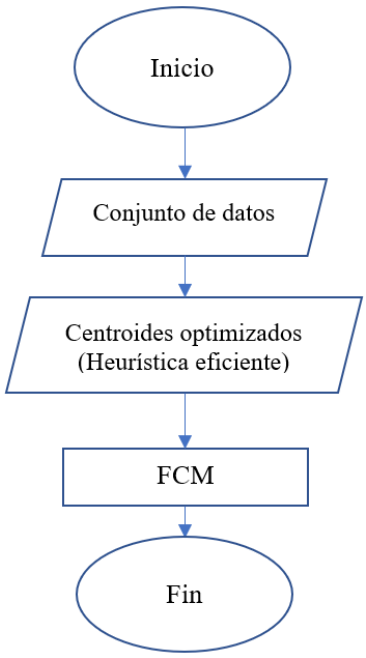


Figura 9. Enfoque de propuesta en la fase de inicialización.

Como se observa en la Figura 9, se parte de un conjunto de datos, del cual se seleccionarán centroides iniciales para realizar la agrupación con Fuzzy *C*-Means. Esto se logrará con el desarrollo de una heurística optimizada, la cual realizará un pre-procesamiento de los datos, y generará centroides optimizados que serán usados por Fuzzy *C*-Means en la fase de inicialización.

De la literatura especializada, el artículo que más se acerca a los objetivos de la investigación es la publicación de (Pérez, J. et al., 2022) en donde se describe una mejora al algoritmo Fuzzy *C*-Means. Esto se logra con dos variantes de K-Means denominadas K++ (Arthur, D., 2007) y O-K-Means (Pérez, J. et al., 2018), se realiza la transformación de los centroides finales de los algoritmos a matriz de pertenencia y esta última es usada por el algoritmo Fuzzy *C*-Means.

Se sabe que la eficiencia del algoritmo Fuzzy *C*-Means depende, entre otros factores, de las estrategias de inicialización y convergencia. Con base a las publicaciones (Pérez, J. et al., 2022) y (Pérez, J. et al., 2024) se desarrolló una heurística híbrida, la cual integra los aportes de ambas publicaciones, esta heurística se denominó Hybrid Pareto Fuzzy C-Means (HPFCM). En la primera, se realiza un aporte a la mejorar los parámetros iniciales del algoritmo Fuzzy *C*-Means, en particular los grados de membresías; la segunda, se describe un aporte a un nuevo criterio de paro para el algoritmo Fuzzy *C*-Means.

Algorithm 2: HPFCM

Input: *X, c, m,* ε*, T*
Output: *V, U*
1 **Initialization:**
2 *εok*: = Threshold value for determining O-K-Means convergence;
3 Assign the value for *c*;
4 Function K++ (*X*, *c*):
5 Return *V'*;
6 Function O-K-Means (*X*, *V´*, *εok*, *c*):
7 Return *V''*;
8 Function *S* (*V''*):
9 Return *U*;
10 Determine the value of the threshold ε to determine the convergence of the algorithm;
11 Assign the value for *m*;
12 *t*: = 0;
13 **Calculate centroids:**
14 Calculate the centroids using Exp. (5);
15 **Classification:**
16 Update and calculate the membership matrix using Exp. (4);
17 **Convergence:**
18 **If** $\Delta J_m^{(t)} \leq \varepsilon$:
19 Stop the algorithm;
20 **Otherwise**:
21 $U^{(t)} := U^{(t+1)}$ y $t := t + 1$;
22 Go to **Classification**
23 **End of algorithm**

El algoritmo consta de cuatro etapas, pero la integración de las heurísticas se enfoca en dos etapas que son inicialización y convergencia. El primero realiza tres funciones: La función K++, tiene como parámetro de entrada el conjunto de datos *X* y el número de grupos; esta función tiene como parámetro de salida el conjunto de centroides identificados con la variable *V'*. La función OK-Means, tiene como parámetro de entrada el conjunto de datos *X*, el conjunto de centroides *V'* el valor del umbral *εok* y el número de grupos; esta función tiene como parámetro de salida el conjunto de centroides identificados con la variable *V''*. La función *S*, tiene como parámetro de entrada el conjunto centroides identificados con la variable *V''*; esta función tiene como parámetro de salida una matriz de membresía calculada a partir de *V''*. La segunda y tercera fase, llamadas cálculo de centroides y clasificación, respectivamente, son típicas del Fuzzy *C*-Means. La fase de convergencia, realiza la evaluación del indicador de paro de P-FCM, en donde se calcula la diferencia en porcentaje de la función objetivo en dos iteraciones consecutivas y el valor se compara con un umbral definido. Este indicador se muestra en la Expresión 23 y se evalúa en la línea 18 del algoritmo 2.

Capítulo 5

5. Experimentación y validación de resultados

En este capítulo se describen los experimentos realizados para validar las heurísticas propuestas en el presente proyecto de investigación. Se realizaron dos planes de prueba para validar cada heurística, la primera, evalúa la heurística P-FCM; la segunda, evalúa la heurística HPFCM. Cada plan de pruebas se realizó con diferentes conjuntos de datos reales y algoritmos del estado del arte.

5.1 Plan de pruebas para la heurística P-FCM

5.1.1 Introducción

En esta sección, se presentan las pruebas experimentales a partir de la heurística desarrollada P-FCM para reducir la complejidad del algoritmo Fuzzy *C*-Means. El propósito es evaluar la reducción de la complejidad temporal del algoritmo Fuzzy *C*-Means a través de la fase de convergencia con la heurística P-FCM. Se resolvieron conjuntos de datos reales y sintéticos con el algoritmo Fuzzy *C*-Means con valor de umbral $\varepsilon = 0.01$, ya que este valor es el más usado en la literatura. También se resolvieron los conjuntos de datos con el algoritmo P-FCM con los siguientes valores de umbral propuestos: el valor de $\varepsilon = 0.9$, prioriza una reducción en la cantidad de iteraciones; el valor de $\varepsilon = 0.06$ prioriza un mejor porcentaje acumulado de la función objetivo; y el valor $\varepsilon = 0.22$, es un equilibrio en la reducción de iteraciones y el porcentaje acumulado de la función objetivo. Para los conjuntos de datos sintéticos se usó un valor de $C = 10$, y para los conjuntos de datos reales un valor de $C = 50$, este valor es el más alto reportado en la literatura (Vimala, S. y Vivekanandan, K., 2019). Los algoritmos fueron implementados en lenguaje C, se usó el compilador GCC 7.4.0. Los experimentos fueron ejecutados en equipo de cómputo Acer Nitro 5 con sistema operativo Windows 11 procesador

Intel ® Core ™ i7-11800H a 2.30 GHz, con 16 GB de memoria RAM, 512 GB SSD y 1 TB HDD.

5.1.2 Conjuntos de datos

Los conjuntos de datos de prueba que se utilizaron para evaluar la heurística P-FCM son reales y sintéticos. En la Tabla 8, se muestran los conjuntos de datos reales obtenidos del UCI *Machine Learning Repository* (Markelle, K., et al., 2023). La estructura de la tabla es la siguiente: la columna uno, muestra el nombre del conjunto de datos; la columna dos, muestra la cantidad de clases que tiene el conjunto de datos; la columna tres, muestra si es un conjunto de datos real o sintético; la columna cuatro, muestra la cantidad de filas; la columna cinco, muestra la cantidad de atributos; la columna seis, muestra el producto de la columna cuatro y cinco.

Tabla 8. Conjuntos de datos usados en el primer plan de pruebas.

Nombre	Clases	Tipo	*n*	*d*	*n*d*
ABALONE	3	Real	4,177	7	29,239
WINE	11	Real	4,898	11	53,878
URBAN	1	Real	360,177	2	720,354
S3_2	2	Sintético	30	2	60
S6_2	4	Sintético	63	2	126
S8_2	3	Sintético	84	2	168
S10_2	1	Sintético	100	2	200

El conjunto de datos *Abalone*, contiene datos de mediciones físicas. El objetivo es determinar las edades de los abulones. La edad del abulón se determina cortando la concha a través del cono, tiñéndola y contando el número de anillos a través de un microscopio.

El conjunto de datos *Wine*, contiene datos de un análisis químico de vinos que fueron cultivados en Italia, pero de distintos cultivares. El conjunto de datos contiene valores del ácido málico, magnesio entre otras. El objetivo es diferenciar los vinos de otros mediante sus características.

El conjunto de datos *Urban*, contiene coordenadas (longitud y latitud) de 360,177 accidentes de tráfico ocurridos en zonas urbanas de Gran Bretaña, y etiquetados según el centro urbano donde ocurrieron (469 etiquetas posibles).

Los conjuntos de datos sintéticos contienen valores entre 1 – 10. Estos conjuntos de datos fueron creados usando la herramienta *R* con la finalidad de crear diferentes formas en una distribución de dos dimensiones.

5.1.3 Algoritmo de comparación

Algoritmo Fuzzy C-Means

El algoritmo Fuzzy C-Means es un método de agrupamiento fundamentado en la teoría de conjuntos difusos. El algoritmo Fuzzy C-Means permite que cada dato pertenezca a más de un grupo asignando grados de pertenencia con valores entre 0 y 1. Su descripción se encuentra en la sección 2.3. El pseudocódigo 3, muestra el algoritmo Fuzzy C-Means.

Algoritmo 3: Fuzzy C-Means

Input: dataset X, c, m, ε, T
Output: V, U

Initialization:
  t: = 0;
  ε: = 0.01
  $U^{(t)}$: = $\{u_{11}, \ldots, u_{ij}\}$; is randomly generated
Calculate centroids:
  Calculate the centroids using Exp. (5);
Calculate membership matrix:
  Update and calculate the membership matrix using Exp. (4);
Convergence:
  If $\Delta U^{(t)} < \varepsilon$ or $t < T$:
    Stop the algorithm;
  Otherwise:
    $U^{(t)}$: = $U^{(t+1)}$ and t: = t + 1;
  Go to **Calculate centroids**
End of algorithm

5.1.4 Resultados del plan de pruebas con la heurística P-FCM

En esta sección, se muestra el análisis de resultados con conjuntos de datos reales y sintéticos. Se realizaron 5 ejecuciones con cada conjunto de datos real, En la Figura 10, se muestra el comportamiento del porcentaje acumulado de la función objetivo a través de las iteraciones con el conjunto de datos *Abalone*.

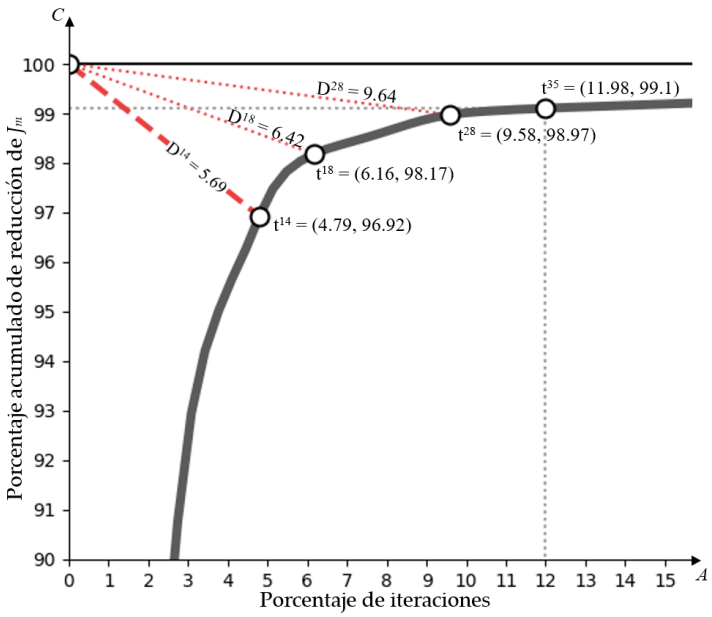


Figura 10. Optimidad con el conjunto de datos *Abalone*.

En la Figura 10, se muestra la relación entre el esfuerzo computacional y la calidad de solución del algoritmo P-FCM con el conjunto de datos *Abalone*. El punto t^{18}, representa el esfuerzo y la calidad de solución en la iteración 18. El algoritmo P-FCM logró realizar solo el 6.16% del total de iteraciones que realiza el algoritmo FCM. Además, se obtuvo una pérdida de calidad poco significativa de 1.83%, estos resultados se obtienen usando un valor de $\varepsilon = 0.9$ como umbral de paro. Por otro lado, usar un valor de $\varepsilon = 0.22$, se logra obtener el resultado del punto t^{28}, aquí la convergencia se da en la iteración 18, lo que representa el 6.16% del total de iteraciones que hace FCM, también se logra alcanzar un porcentaje acumulado de la función objetivo del 98.97%. El punto t^{35} representa una convergencia en la iteración 35, esta iteración representa el 11.98% del total de iteraciones que realiza FCM, de igual manera se alcanza un porcentaje acumulado de la función objetivo del 99.1%, estos valores se obtienen usando un valor de umbral de $\varepsilon = 0.06$. En la Tabla 9, se detallan resultados de interés al resolver el conjunto de datos *Abalone*, con P-FCM.

Tabla 9. Resultados de interés con *Abalone*.

Iteración (t)	$J_m^{(t)}$	A^t	B^t	C^t	D^t	$\Delta J_m^{(t)}$	$\Delta U^{(t)}$
1	28.20	0.34	--	--	--	--	0.98
--	--	--	--	--	--	--	--
8	7.01	2.73	3.09	90.75	9.63	9.36	0.61
--	--	--	--	--	--	--	--
13	5.71	4.45	0.59	96.25	5.81	2.36	0.71
14	**5.56**	**4.79**	**0.66**	**96.92**	**5.69**	**2.71**	**0.43**
--	--	--	--	--	--	--	--
18	5.26	**6.16**	0.14	**98.17**	6.42	**0.64**	0.18
--	--	--	--	--	--	--	--
28	5.08	**9.58**	0.04	**98.97**	9.64	**0.21**	0.07
--	--	--	--	--	--	--	--
35	5.05	**11.98**	0.01	**99.1**	12.01	**0.05**	0.05
--	--	--	--	--	--	--	--
292	4.84	100	0.0005	100	100	0.0024	0.0094

En la Tabla 9, se muestran los resultados con el conjunto de datos *Abalone*. Se destacan los resultados obtenidos con los valores de ε = 0.9, 0.22, 0.06 con los valores de iteración (t) = 18, 28 y 35 respectivamente. En la Figura 11, se muestra el esfuerzo computacional y la calidad de solución del P-FCM con el conjunto de datos *Wine*.

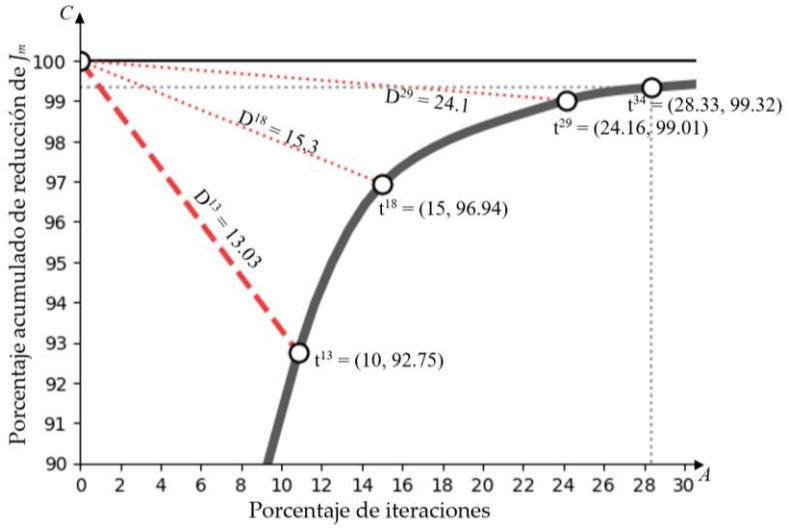


Figura 11. Optimidad con el conjunto de datos *Wine*.

En la Tabla 10, se muestran los resultados al resolver el conjunto de datos *Wine*. Se destacan los puntos de interés (optimalidad de Pareto) mostrados en la Figura 11, que son t^{18}, t^{29} y t^{34} el momento en que se alcanzó el porcentaje acumulado 99% de la función objetivo.

Tabla 10. Resultados de interés con *Wine*.

Iteración (t)	$J_m^{(t)}$	A^t	B^t	C^t	D^t	$\Delta J_m^{(t)}$	$\Delta U^{(t)}$
1	207,529.78	0.83	--	--	--		0.25
--	--	--	--	--	--	--	--
13	**84,076.47**	**10.83**	**1.51**	**92.75**	**13.03**	**2.33**	**0.38**
--	--	--	--	--	--	--	--
18	5.26	**15**	0.49	**96.94**	15.03	**0.83**	0.23
--	--	--	--	--	--	--	--
29	75,739.98	**24.16**	0.12	**99.01**	24.18	**0.21**	0.11
--	--	--	--	--	--	--	--
34	75,321.91	**28.33**	0.03	**99.32**	28.34	**0.05**	0.07
--	--	--	--	--	--	--	--
120	74,428.39	100	0.0002	100	100	0.0003	0.0098

En la Tabla 10, se muestran los resultados de P-FCM con el conjunto de datos *Wine*. Es destacable que usar un valor de $\varepsilon = 0.9$, realiza un total de 18 iteraciones que representa 15% del total de iteraciones, con una pérdida de calidad poco significativa de 2.06%; con un valor de $\varepsilon = 0.22$, se realizan solo 29 iteraciones que representa el 24.16% del total de iteraciones, con una pérdida de calidad de 0.99%; y un valor de $\varepsilon = 0.06$, genera solo 34 iteraciones que representa el 28.33% del total de iteraciones, con una pérdida de calidad de 0.68%. En la Figura 12, se muestra el comportamiento del porcentaje acumulado de la función objetivo a través de las iteraciones con el conjunto de datos *Urban*.

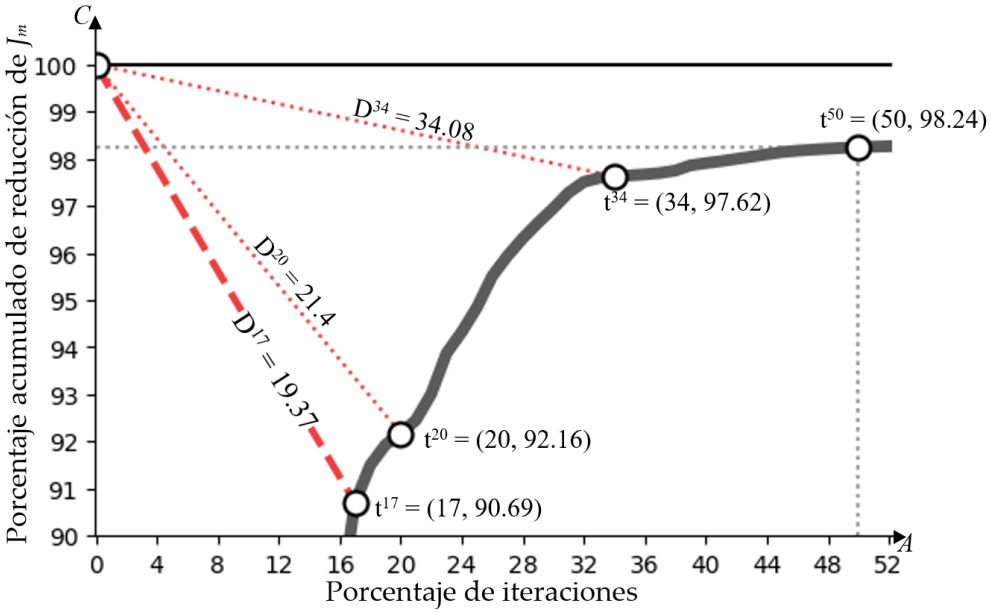


Figura 12. Optimidad con el conjunto de datos *Urban*.

En la Tabla 11, se muestran algunos resultados de interés al resolver el conjunto de datos *Urban*. Se destacan los puntos de interés (t^{20}, t^{34} y t^{50}) mostrados en la Figura 12.

Tabla 11. Resultados de interés con *Urban*.

Iteración (t)	$J_m^{(t)}$	A^t	B^t	C^t	D^t	$\Delta J_m^{(t)}$	$\Delta U^{(t)}$
1	26,283.06	1	--	--	--	--	0.56
--	--	--	--	--	--	--	--
17	**7,637.21**	**17**	**2.16**	**90.69**	**19.37**	**5.51**	**0.96**
--	--	--	--	--	--	--	--
20	7,335.26	**20**	0.23	**92.16**	21.47	**0.66**	0.22
--	--	--	--	--	--	--	--
34	6,213.83	**34**	0.03	**97.62**	34.08	**0.1**	0.29
--	--	--	--	--	--	--	--
50	6,086.22	**50**	0.01	**98.24**	50.03	**0.04**	0.16
--	--	--	--	--	--	--	--
100	5,724.92	100	0.00005	100	100	0.0001	0.0087

En la Tabla 11, se muestran los resultados con el conjunto de datos *Urban*. Es destacable que asignar un valor de $\varepsilon = 0.9$, se genera solo 20 iteraciones, esto representa una reducción del 80% de las iteraciones que realiza el algoritmo FCM, con una pérdida de calidad de 7.84%; un valor de $\varepsilon = 0.22$, reduce el 66% de las iteraciones, con una pérdida de calidad de 2.38%; y un valor de $\varepsilon = 0.06$, reduce el 50% de las iteraciones, con una pérdida de calidad de 1.76%. En la Figura 13, se muestra el esfuerzo computacional y la calidad de solución del P-FCM al resolver el conjunto de datos sintético S3_2.

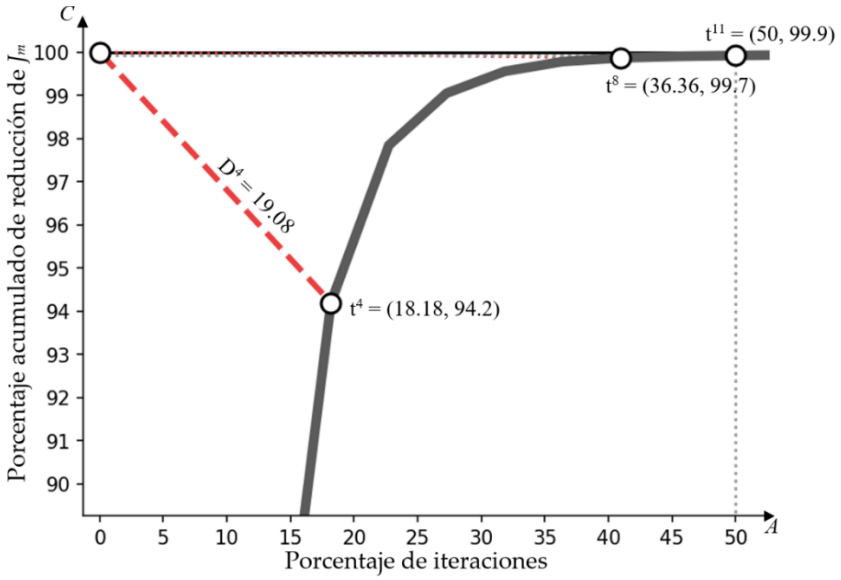


Figura 13. Optimidad con el conjunto de datos *S3_2*.

En la Tabla 12, se muestran algunos resultados de interés al resolver el conjunto de datos *S*3_2. Se destacan los puntos de interés (t^8 y t^{11}) mostrados en la Figura 13.

Tabla 12. Resultados de interés con el conjunto de datos *S3_2*.

Iteración (t)	$J_m^{(t)}$	A^t	B^t	C^t	D^t	$\Delta J_m^{(t)}$	$\Delta U^{(t)}$
1	54.28	4.54	--	--	--	--	0.92
--	--	--	--	--	--	--	--
4	**19.71**	**18.18**	**10.71**	**94.20**	**19.08**	**16.61**	**0.36**
--	--	--	--	--	--	--	--
8	17.67	**36.36**	0.22	**99.78**	36.36	**0.47**	0.07
--	--	--	--	--	--	--	--
11	17.61	**50**	0.02	**99.92**	50	**0.04**	0.02
--	--	--	--	--	--	--	--
22	17.59	100	0.0017	100	100	0.0036	0.0092

En la Tabla 12, se muestran los resultados con el conjunto de datos *S3_2*. Es destacable algunos puntos de interés que son t^8 y t^{11}, los cuales se encuentran cerca del 100% del porcentaje acumulado de la función objetivo. Los puntos t^8 y t^{11} fueron obtenidos al usar valores de ε = 0.9 y 0.06 respectivamente, se destacan para mostrar que alcanzaron una calidad de solución mayor al 99%. Además, redujeron la cantidad de iteraciones hasta un 50% del total de iteraciones que realiza el algoritmo FCM. En la Figura 14, se muestra el comportamiento del porcentaje acumulado de la función objetivo a través de las iteraciones con el conjunto de datos *S*6_2.

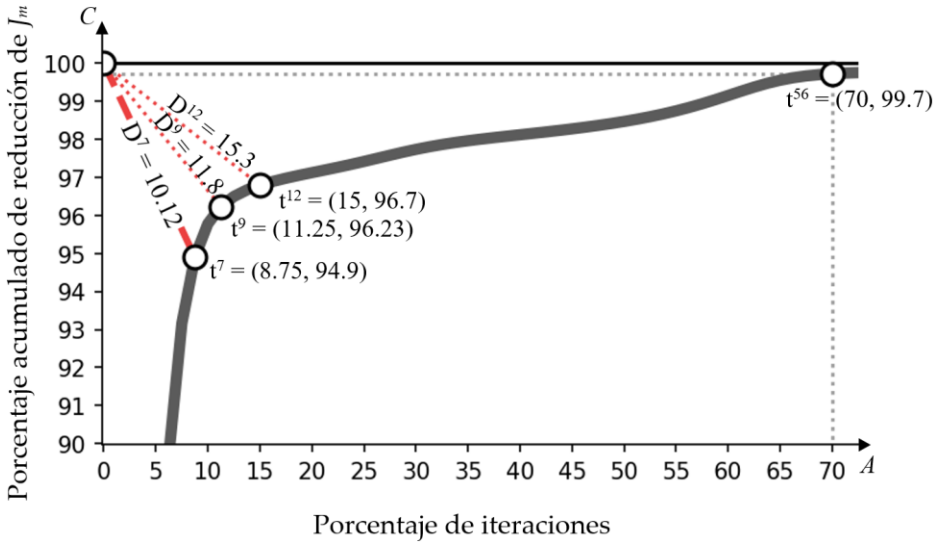


Figura 14. Optimidad con el conjunto de datos *S6_2*.

En la Tabla 13, se muestran algunos resultados de interés al resolver el conjunto de datos sintético S6_2. Se destacan los puntos de interes (t^9, t^{12} y t^{56}) mostrados en la Figura 14.

Tabla 13. Resultados de interés con el conjunto de datos *S6_2*.

Iteración (t)	$J_m^{(t)}$	A^t	B^t	C^t	D^t	$\Delta J_m^{(t)}$	$\Delta U^{(t)}$
1	37.59	1.25	--	--	--	--	0.88
--	--	--	--	--	--	--	--
7	**14.44**	**8.75**	**1.74**	**94.90**	**10.12**	**2.86**	**0.18**
--	--	--	--	--	--	--	--
9	14.11	**11.25**	0.45	**96.23**	11.86	**0.77**	0.06
--	--	--	--	--	--	--	--
12	13.98	**15**	0.12	**96.79**	15.33	**0.21**	0.04
--	--	--	--	--	--	--	--
56	13.26	**70**	0.02	**99.72**	70	**0.04**	0.01
--	--	--	--	--	--	--	--
80	13.20	100	0.0051	100	100	0.0094	0.0088

En la Tabla 13, se muestran los resultados con el conjunto de datos *S6_2*. Es destacable que usar ε = 0.9, generó solo nueve iteraciones lo que representa el 11.25% de iteraciones que realiza FCM, con una pérdida de calidad de 3.77%. Por otro lado, usar un valor de ε = 0.22 con el algoritmo P-FCM se realizan solo 12 iteraciones, lo que representa el 15% del total que realiza FCM, con un porcentaje acumulado de la función objetivo de 96.79%. Usar un valor de ε = 0.06, generó 56 iteraciones lo que representa el 70% de las iteraciones que realiza FCM, pero con una pérdida de calidad poco significativa de 0.3%. En la Figura 15, se muestra el comportamiento del porcentaje acumulado de la función objetivo a través de las iteraciones con el conjunto de datos *S*8_2.

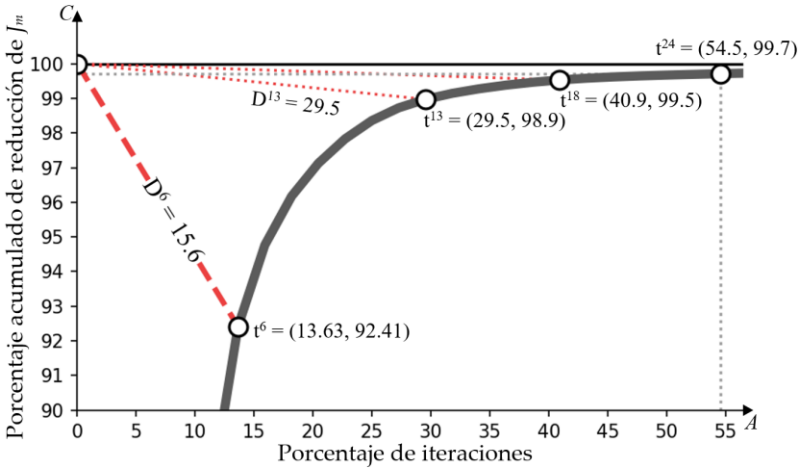


Figura 15. Optimidad con el conjunto de datos *S8_2*.

En la Tabla 14, se muestran algunos resultados de interés al resolver el conjunto de datos sintético S8_2. Se destacan los puntos de interes (t^{13}, t^{18} y t^{24}) mostrados en la Figura 15.

Tabla 14. Resultados de interés con el conjunto de datos *S8_2*.

Iteración (t)	$J_m^{(t)}$	A^t	B^t	C^t	D^t	$\Delta J_m^{(t)}$	$\Delta U^{(t)}$
1	81.88	2.27	--	--	--	--	0.93
6	**24.84**	**13.63**	**4.73**	**92.41**	**15.60**	**10.52**	**0.30**
--	--	--	--	--	--	--	--
13	20.79	**29.54**	0.24	**98.97**	29.56	0.72	0.05
--	--	--	--	--	--	--	--
18	20.44	**40.9**	0.06	**99.52**	40.91	**0.19**	0.03
--	--	--	--	--	--	--	--
24	20.33	**54.54**	0.01	**99.71**	54.54	0.05	0.02
--	--	--	--	--	--	--	--
44	20.15	100	0.0017	100	100	0.0052	0.0087

En la Tabla 14, se muestran los resultados con el conjunto de datos *S*8_2. Es destacable que los puntos t^{13}, t^{18} y t^{24}, obtienen un porcentaje acumulado de la función objetivo mayor al 90%. Por ejemplo, el punto t^{13} muestra un porcentaje acumulado del 98.9%, este acumulado se obtuvo al usar un valor de ε = 0.9. Además, generó solo 13 iteraciones, representando el 29.5% del total que genero FCM. En la Figura 16, se muestra el comportamiento del porcentaje acumulado de la función objetivo a través de las iteraciones con el conjunto de datos *S*10_2.

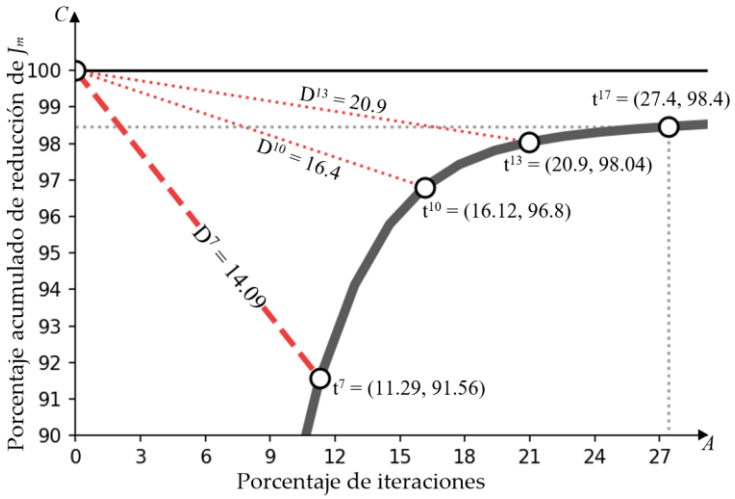


Figura 16. Optimidad con el conjunto de datos *S10_2*.

En la Tabla 15, se muestran algunos resultados de interés al resolver el conjunto de datos sintético S10_2. Se destaca la optimalidad de pareto con los puntos de interés (t^{10}, t^{13} y t^{17}) mostrados en la Figura 16.

Tabla 15. Resultados de interés con el conjunto de datos *S10_2*.

Iteración (t)	$J_m^{(t)}$	A^t	B^t	C^t	D^t	$\Delta J_m^{(t)}$	$\Delta U^{(t)}$
1	149.82	1.61	--	--	--	--	0.93
--	--	--	--	--	--	--	--
7	**86.34**	**11.29**	**3.89**	**91.56**	**14.09**	**3.03**	**0.14**
--	--	--	--	--	--	--	--
10	82.71	**16.12**	1.01	**96.80**	16.44	**0.84**	0.08
--	--	--	--	--	--	--	--
13	81.85	**20.96**	0.23	**98.04**	21.05	**0.20**	0.04
--	--	--	--	--	--	--	--
17	81.56	**27.41**	0.06	**98.46**	27.46	**0.05**	0.01
--	--	--	--	--	--	--	--
62	80.49	100	0.01	100	100	0.0123	0.0097

En la Tabla 15, se muestran los resultados con el conjunto de datos *S10_2*. Es destacable que, usar cualquier valor de ε se alcanza un porcentaje acumulado de la función objetivo mayor al 90%. En particular, usar un valor de $\varepsilon = 0.06$ (punto t^{17}) redujo el 72.6% de las iteraciones que realiza el algoritmo FCM, con una pérdida de calidad poco significativa de 1.6%.

5.1.5 Análisis de resultados

Con la heurística propuesta P-FCM, se logró reducir la complejidad del algoritmo Fuzzy *C*-Means a través de las iteraciones, con pérdidas de calidad poco significativas. De manera general, se encontró que el algoritmo tiene un comportamiento ineficiente en más de la mitad de las iteraciones. Esto es, para ir reduciendo el valor de la función objetivo Fuzzy *C*-Means cada vez va requiriendo de un mayor número de iteraciones. Así, por ejemplo, en una ejecución del algoritmo, éste alcanzó el 96.92% de calidad en la iteración 14, el 99% de calidad en la iteración 29 y el 100% de calidad en la iteración 292. En este ejemplo, si el algoritmo hubiese parado en la iteración 29 la calidad solamente se hubiera reducido un 1%, sin embargo, las iteraciones un 90.07%. Como parte del análisis de los resultados

experimentales, se observó de manera general que el valor de 0.01 está sobredimensionado, e incide negativamente en la eficiencia del algoritmo. En este sentido, propusieron nuevos valores de umbral, de propósito general. Al usar un valor de umbral del 0.22, se obtiene un valor aproximado al 98% de calidad en la solución y una reducción aproximada del 94.18% de iteraciones; un valor de umbral 0.06 obtiene el 99% de calidad con una reducción del 90.07% de iteraciones.

5.2 Plan de pruebas para la heurística HPFCM

5.2.1 Introducción

En esta sección, se presentan las pruebas experimentales a partir de la heurística desarrollada HPFCM para reducir la complejidad del algoritmo Fuzzy *C*-Means. El propósito de este plan de pruebas, consiste en evaluar la reducción de la complejidad temporal del algoritmo Fuzzy *C*-Means a través de las fases de inicialización y convergencia con la heurística HPFCM. El plan de pruebas consta de conjuntos de datos reales que se solucionaron con los algoritmos Fuzzy *C*-Means, HOFCM, InoFrep y NFCM. Los algoritmos fueron implementados en lenguaje C, se usó el compilador GCC 7.4.0. Los experimentos fueron ejecutados en equipo de cómputo Acer Nitro 5 con sistema operativo Windows 11 procesador Intel ® Core ™ i7-11800H a 2.30 GHz, con 16 GB de memoria RAM, 512 GB SSD y 1 TB HDD.

5.2.2 Conjuntos de datos de prueba

Los conjuntos de datos utilizados fueron reales. Los conjuntos de datos reales fueron obtenidos del *UCI Machine Learning Repository* (Markelle, K., et al., 2023). La Tabla 16, muestra las características de los conjuntos de datos. La estructura de la tabla es la siguiente: la columna uno, muestra el nombre del conjunto de datos; la columna dos, muestra la cantidad de clases que tiene el conjunto de datos; la columna tres, muestra si es un conjunto de datos real; la columna cuatro, muestra la cantidad de filas; la columna cinco, muestra la cantidad de atributos; la columna seis, muestra el producto de la columna cuatro y cinco.

Tabla 16. Conjuntos de datos usados en el segundo plan de pruebas.

Nombre	Clases	Tipo	*n*	*d*	*n*d*
WDBC	2	Real	569	30	17,070
ABALONE	3	Real	4,177	7	29,239
SPAM	2	Real	4,601	57	262,257

URBAN	1	Real	360,177	2	720,354
Emociones	4	Real	749	3	2,247

El conjunto de datos *WDBC*, es un conjunto de datos real, contiene datos agrupados de informes que se realizaron en una clínica de manera periódica que va desde enero de 1989 a noviembre de 1991. Se busca evaluar si un tumor es maligno o benigno.

El conjunto de datos *SPAM*, es un conjunto de datos real, en los atributos hay valores que indican si una palabra o carácter es frecuente en el correo. Se busca determinar si los correos electrónicos son spam o no. El conjunto de datos *Abalone* y *Urban* se describen en la sección 5.1.2.

El conjunto de datos Emociones, contiene datos para el reconocimiento de emociones y se describe en la publicación de (Colunga, A. *et al*., 2025). Los autores desarrollaron el conjunto de datos a partir de un experimento controlado para la recolección señales fisiológicas y multimedia, estas señales se asocian con cuatro estados emocionales (calma, enojo, felicidad y tristeza).

5.2.3 Algoritmos de comparación

Cada conjunto de datos se resolvió con los algoritmos Fuzzy *C*-Means, HOFCM, InoFrep y NFCM. El algoritmo Fuzzy *C*-Means, se implementó para que los resultados puedan ser comparados con el método estándar; los algoritmos HOFCM, InoFrep y NFCM se implementaron por tener un enfoque en la etapa de inicialización de Fuzzy *C*-Means. Se realizaron varias ejecuciones con diferentes configuraciones, tales como, diferentes valores de umbral. En total se realizaron 100 ejecuciones de los algoritmos.

Algoritmo HOFCM

El algoritmo HOFCM es una extensión del algoritmo Fuzzy *C*-Means que incorpora dos mejoras del algoritmo K-Means y se describe en (Pérez, J. et al., 2022). Las mejoras K++ y OK-Means son utilizadas para optimizar los parámetros iniciales del algoritmo Fuzzy *C*-Means. Estos algoritmos proporcionan centroides iniciales, pero mediante una función de transformación *S*, los grados de pertenencia son proporcionados a Fuzzy *C*-Means. El pseudocódigo 4 muestra el algoritmo HOFCM.

Algoritmo 4: HOFCM

Initialization:
  X: $\{x_1, \ldots, x_n\}$;
  V: $\{v_1, \ldots, v_n\}$;
  εok: = Threshold value for determining O-K-Means convergence;
  Assign the value for c;
  Function K++ (X, c):
    Return V';
  Function O-K-Means (X, $V´$, εok, c):
    Return V'';
  Function S (V''):
    Return U;
  Determine the value of the threshold ε to determine the convergence of the algorithm;
  Assign the value for m;
  t: = 0;
Calculate centroids:
  Calculate the centroids using Exp. (5);
Classification:
  Update and calculate the membership matrix using Exp. (4);
Convergence:
  If $[\text{abs}(\mu_{ij}^{(t)} - \mu_{ij}^{(t)})] \leq \varepsilon$:
    Stop the algorithm;
  Otherwise:
    $U^{(t)} := U^{(t+1)}$ y $t := t + 1$;
  Go to **Classification**
End of algorithm

Algoritmo InoFrep

El algoritmo InoFrep es un método de inicialización para optimizar los centroides iniciales para el algoritmo Fuzzy *C*-Means, este algoritmo se describe en (Cebeci, Z. y Cebeci, C., 2020). El algoritmo determina la distribución de frecuencias en los datos, identifica los picos en cada dimensión del conjunto de datos y los clasifica como posibles candidatos a centroides iniciales. El pseudocódigo 5 muestra el algoritmo InoFrep.

Algoritmo 5: InoFrep

Input: dataset X, c, m, ε, T
Output: V, U

Initialization:
    t: = 0;
    ε: = 0.01
    V: = {};
    For j from 1 to d:
        values: = Extract column j from X;
        peaks: = Find peaks in *values* using the finite difference method;
        If counter_*peaks* $\geq c$:
            Order *peaks* from highest to lowest;
            Select the first c peaks;
            $V[:, j]$: = *peaks* [1, …, c];
        Otherwise:
            Fill $V[:, j]$ i-th random values from the range of *values*;
    End For
Calculate membership matrix:
    Update and calculate the membership matrix using Exp. (4);
Calculate centroids:
    Calculate the centroids using Exp. (5);
Convergence:
    If $\Delta U^{(t)} < \varepsilon$ or $t < T$:
        Stop the algorithm;
    Otherwise:
        $U^{(t)}$: = $U^{(t+1)}$ and t: = t + 1;
    Go to **Calculate membership matrix**
End of algorithm

Algoritmo NFCM

El algoritmo NFCM, es un algoritmo enfocado en la fase de inicialización de Fuzzy *C*-Means y se basa de una distribución de probabilidad mejorada. En lugar de usar una distribución D^2-weighting como el algoritmo FCM++, utiliza una distribución μ^2-weighting, que tiene en cuenta la contribución de cada punto a la función objetivo de Fuzzy *C*-Means. Este algoritmo se describe en (Qian, L. *et al*., 2021) y se muestra en el pseudocódigo 6.

Algoritmo 6: NFCM

Input: dataset X, c, m, ε, T
Output: V, U

Initialization:
  t: = 0;
  ε: = 0.01
  V:= $V \cup \{c_1\}$; is randomly generated
  For i from 2 to c:
    Select the i-th center v_i, selecting $c_i = x' \in X - C$ probability ($\varphi(x',C)$ / $\varphi(X,C)$);
    V:= $V \cup \{c_i\}$;
  End For
Calculate centroids:
  Calculate the centroids using Exp. (5);
Calculate membership matrix:
  Update and calculate the membership matrix using Exp. (4);
Convergence:
  If $\Delta U^{(t)} < \varepsilon$ or $t < T$:
    Stop the algorithm;
  Otherwise:
    $U^{(t)}$: = $U^{(t+1)}$ and t: = t + 1;
  Go to **Calculate centroids**
End of algorithm

5.2.4 Resultados del plan de pruebas con la heurística HPFCM

En esta sección, se muestra el análisis de resultados con conjuntos de datos reales y sintéticos. Se agrupó cada conjunto de datos real y sintético, incrementando los valores de *C* para evaluar la escalabilidad de la heurítica propuesta, los cuales son [5, 10, 15, 20, 25, 30, 35, 40, 45 y 50]. De acuerdo con la publicación de (Vimala, S. y Vivekanandan, K., 2019), el valor de $C = 50$ es el más alto reportado en la literatura.

Para cuantificar las diferencias entre los resultados de los algoritmos se establecieron dos indicies α y β, en donde α, expresa la reducción del número de iteraciones de HPFCM con respecto a Fuzzy *C*-Means como un porcentaje del valor de Fuzzy *C*-Means, ver detalle

en la expresión 26. El indicador β, expresa la reducción de la función objetivo de HPFCM con respecto a Fuzzy *C*-Means como un porcentaje, ver expresión 27.

$$\alpha = 100 - \left(100 * \frac{t\_averageHPFCM}{t\_averageFCM}\right) \tag{26}$$

$$\beta = 100 - \left(100 * \frac{fo\_averageHPFCM}{fo\_averageFCM}\right) \tag{27}$$

La Figura 17, consiste de dos gráficas que muestran el resultado de agrupar el conjunto de datos *WDBC*. La primera gráfica, muestra la escalabilidad del algoritmo propuesto frente a los algoritmos comparativos; la segunda, muestra los resultados de resolver el conjunto de datos *WDBC* con 50 grupos. Nótese que, por cada algoritmo se realizaron cinco ejecuciones.

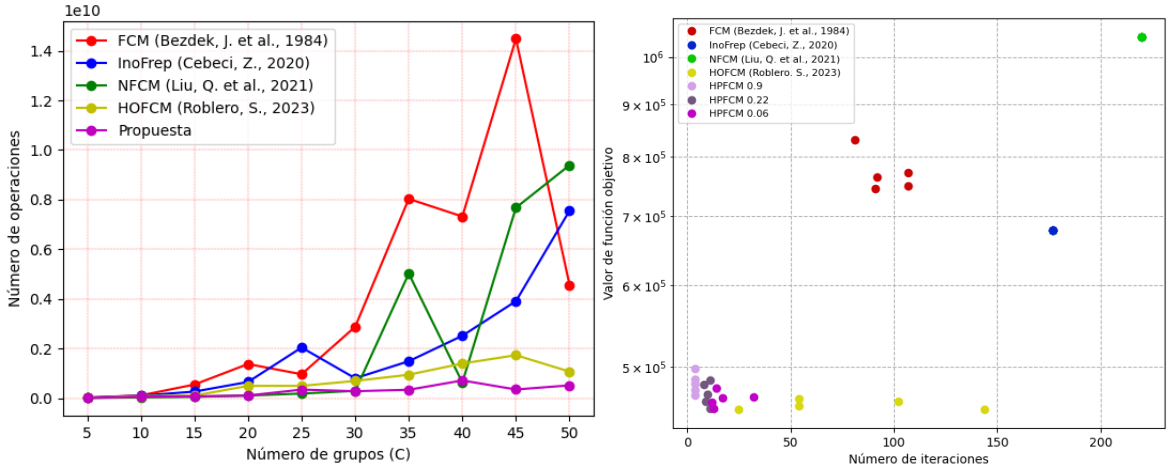


Figura 17. Resultados con el conjunto de datos *WDBC*.

Como se observa en la Figura 17, la primera gráfica muestra que el algoritmo propuesto tiene una tendencia constante al incrementar el valor de *C*, mientras que los otros algoritmos tienen un crecimiento exponencial. Además, en la segunda gráfica, se observa que la heurística propuesta obtuvo menor cantidad de iteraciones y un menor valor de la función objetivo. En la Tabla 17 se muestran algunos resultados cuantitativos con Fuzzy *C*-Means y HPFCM.

Tabla 17. Resultados de Fuzzy *C*-Means y HPFCM con el conjunto de datos *WDBC*.

Ejecución	FCM		HPFCM 0.9		HPFCM 0.06	
	t	FO	t	FO	t	FO
1	107	749,256.98	4	481,344.46	17	466,814.16
2	92	764,026.31	4	486,051.21	14	476,482.08
3	81	831,151.69	4	498,014.16	32	467,050.78
4	107	771,827.15	4	469,648.82	13	455,164.07
5	91	745,695.01	4	475,056.66	12	461,856.56
Promedio	95.60	772,391.43	4.00	482,023.06	17.60	465,473.53

Como se observa en la Tabla 17, Fuzzy *C*-Means obtuvo en promedio, un valor en la función objetivo de 777,391.43 mientras que HPFCM 0.06 un valor de 465,473.53. Como se observa en la Figura 17 (gráfica 2), los resultados de Fuzzy *C*-Means (marcados en rojo), se encuentran con valores altos de función objetivo e iteraciones; los resultados de HPFCM (marcados en morado), se encuentran cerca del punto de origen. Debido a que la separación entre estos resultados es grande, se muestra que HPFCM presentó mejor desempeño que el algoritmo Fuzzy *C*-Means. En el mejor de los casos, se observó que HPFCM 0.9 obtuvo un valor de α = 95.82 y HPFCM 0.06 un valor de β = 39.73. En la Figura 18, se muestran los resultados al resolver el conjunto de datos *Abalone*.

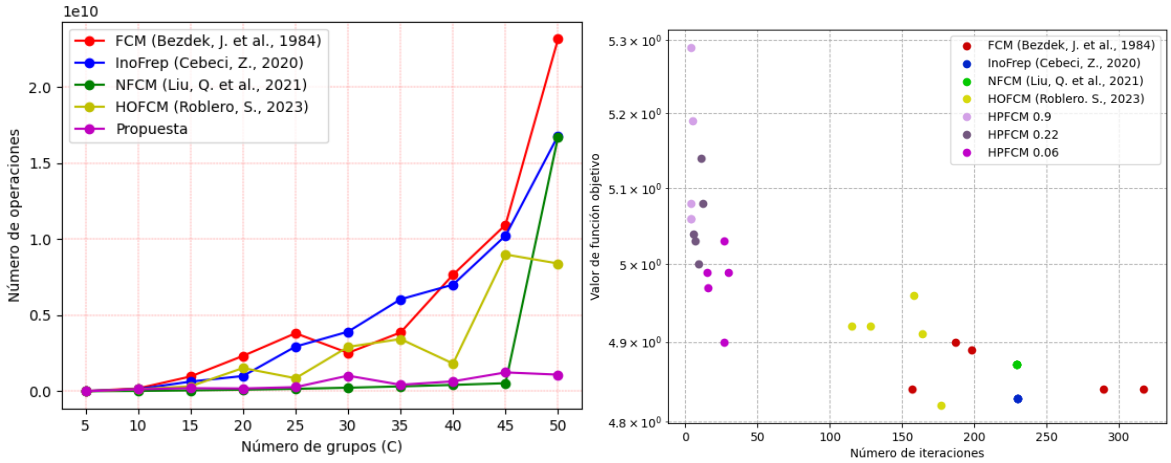


Figura 18. Resultados con el conjunto de datos *Abalone*.

Como se observa en la Figura 18, la primera gráfica muestra que el algoritmo propuesto mantiene una tendencia constante al incrementar el valor de *C*. Nótese que el algoritmo NFCM tiene un comportamiento similar, pero al usar 50 grupos, su complejidad fue mucho mayor que los otros algoritmos. En la segunda gráfica, se observa que la heurística

propuesta obtuvo menor cantidad de iteraciones y un menor valor de la función objetivo. En la Tabla 18, se muestran algunos resultados cuantitativos con Fuzzy *C*-Means y HPFCM.

Tabla 18. Resultados de Fuzzy *C*-Means y HPFCM con el conjunto de datos *Abalone*.

Ejecución	FCM		HPFCM 0.9		HPFCM 0.06	
	t	FO	t	FO	t	FO
1	198	4.89	4	5.29	27	5.03
2	289	4.84	4	5.06	27	4.90
3	157	4.84	4	5.06	15	4.99
4	187	4.90	4	5.08	16	4.97
5	317	4.84	5	5.19	30	4.99
Promedio	229.60	4.86	4.20	5.14	23.00	4.98

Como se observa en la Tabla 18, Fuzzy *C*-Means tuvo menor valor de la función objetivo. Sin embargo, el número de iteraciones fue mucho mayor que HPFCM. En general, Fuzzy *C*-Means obtuvo en promedio un valor de función objetivo de 4.86 y 229.60 iteraciones, mientras que HPFCM 0.06, un valor de 4.98 de función objetivo y 23 iteraciones. En la Figura 18 (gráfica 2), se muestra que los valores de Fuzzy *C*-Means (marcados en rojo) se encuentran alejados del punto origen. Sin embargo, HPFCM 0.06 (marcados en morado) obtuvo resultados cercanos al punto de origen. Debido a que se obtuvieron resultados cercanos al punto (0, 0), se muestra que HPFCM presentó mejor desempeño que el algoritmo Fuzzy *C*-Means. En el mejor de los casos, HPFCM 0.9 obtuvo un valor de α = 98.17 y HPFCM 0.06 obtuvo una reducción de la calidad de β = 5.76. En la Figura 19, se muestran los resultados del conjunto de datos *Spam*.

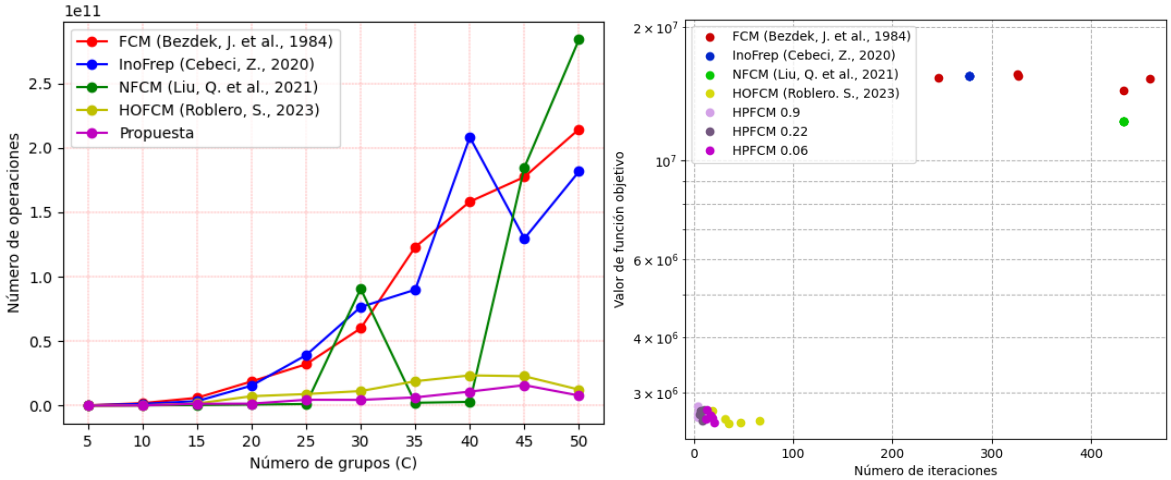


Figura 19. Resultados con el conjunto de datos *Spam*.

Como se observa en la Figura 19, la primera gráfica muestra que el algoritmo propuesto mantiene una tendencia constante al incrementar el valor de *C*. Nótese que el algoritmo NFCM, también tiene un comportamiento similar, pero al usar 30, 45 y 50 grupos, su complejidad fue mucho mayor como los otros algoritmos. En la segunda gráfica, se observa que la heurística propuesta obtuvo menor cantidad de iteraciones y un menor valor de la función objetivo. En la Tabla 19, se muestran algunos resultados cuantitativos con Fuzzy *C*-Means y HPFCM.

Tabla 19. Resultados de Fuzzy *C*-Means y HPFCM con el conjunto de datos *Spam*.

Ejecución	FCM		HPFCM 0.9		HPFCM 0.06	
	t	FO	*t*	FO	*t*	FO
1	327	15,512,347.05	5	2,743,216.21	17	2,672,582.12
2	246	15,346,906.74	4	2,641,992.34	20	2,576,467.98
3	433	14,389,142.01	4	2,800,149.19	13	2,741,455.53
4	326	15,663,485.28	7	2,658,812.92	12	2,615,393.81
5	459	15,254,377.61	4	2,700,152.82	19	2,638,046.71
Promedio	358.2	15,233,251.74	4.80	2,708,864.7	16.20	2,648,789.23

Como se observa en la Tabla 19, Fuzzy *C*-Means generó valores altos de función objetivo e iteraciones. En la Figura 19 (gráfica 2), se muestra los resultados de Fuzzy *C*-Means (marcados en rojo), mientras que los valores de HOFCM y HPFCM (marcados con amarillo y morado) se encuentran cerca al punto origen. En el mejor de los casos, HPFCM 0.9 obtuvo un valor de α = 95.82 y HPFCM 0.06 un valor de β = 39.73. De acuerdo con las observaciones, se muestra que HPFCM tuvo mejor desempeño que el algoritmo Fuzzy *C*-Means. En la Figura 20, se muestran los resultados al resolver el conjunto de datos *Urban*.

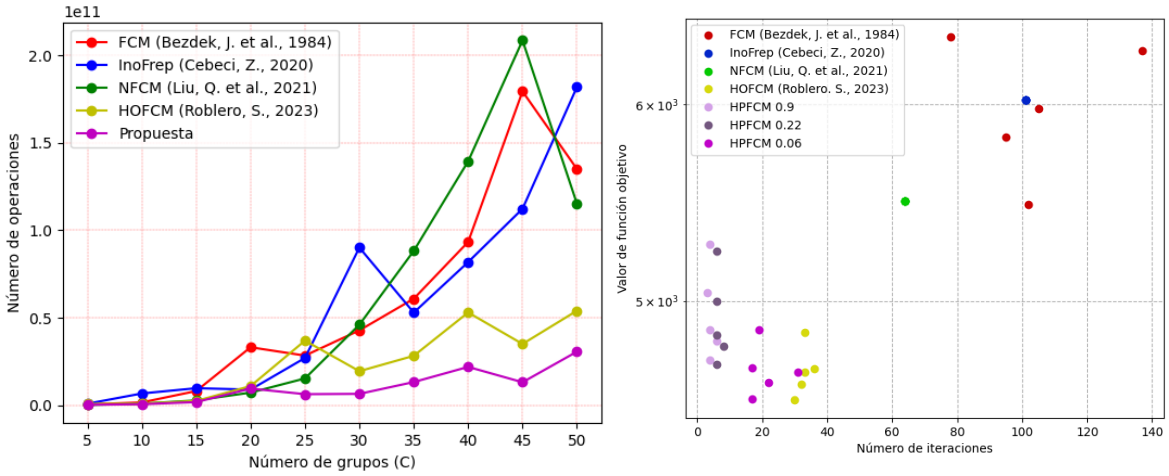


Figura 20. Resultados con el conjunto de datos *Urban*.

Como se observa en la Figura 20, la primera gráfica muestra que el algoritmo propuesto mantiene una tendencia constante al incrementar el valor de *C*. Nótese que, la complejidad de los otros algoritmos es mayor cuando se incrementa el valor de *C*. En la segunda gráfica, se observa que la heurística propuesta obtuvo menor cantidad de iteraciones y un menor valor de la función objetivo. En la Tabla 20, se muestran algunos resultados cuantitativos con Fuzzy *C*-Means y HPFCM.

Tabla 20. Resultados de Fuzzy *C*-Means y HPFCM con el conjunto de datos *Urban*.

Ejecución	FCM		HPFCM 0.9		HPFCM 0.06	
	t	FO	*t*	FO	*t*	FO
1	78	6,386.83	4	4,867.73	22	4,636.03
2	95	5,819.01	4	5,269.01	31	4,681.70
3	137	6,304.24	3	5,038.73	19	4,868.28
4	102	5,468.40	4	4,734.64	17	4,565.79
5	105	5,974.76	6	4,816.80	17	4,701.21
Promedio	103.40	5,990.65	4.20	4,945.38	21.20	4,690.60

Como se observa en la Tabla 20, Fuzzy *C*-Means obtuvo en promedio 5,990.65 en función objetivo y en promedio 103.4 iteraciones, mientras que HPFCM obtuvo resultados bajos en función objetivo e iteraciones. En la Figura 20, se muestra que Fuzzy *C*-Means (marcados en rojo) obtuvo los resultados más altos en función objetivo e iteraciones, mientras que los resultados de HPFCM (marcados en morado) se encuentran cercanos al punto de origen. En el mejor de los casos, HPFCM 0.9 obtuvo un valor de α = 95.94 y HPFCM 0.06 un valor de β = 21.7.

5.2.5 Resultados del plan de pruebas con la heurística HPFCM y Redes Neuronales

En esta sección, se muestra la aplicación de la heurística HPFCM a un caso de estudio. El objetivo es mostrar que, el uso de las heurísticas constituye un método eficiente en la correcta clasificación de datos. Además de reducir la complejidad computacional, la heurística HPFCM, logra que la calidad de agrupamiento se mejore en comparación con otros métodos de clasificación.

Para validar la eficiencia del algoritmo HPFCM, se muestra a continuación los resultados obtenidos con el conjunto de datos de emociones. Este conjunto de datos, se desarrolló a partir de un experimento controlado para la recolección de señales fisiológicas y de multimedia, y se describe en la investigación de (Colunga, A. *et al*., 2025). Contiene tres

atributos (‘*x*’, ‘*y*’, ‘*z*’), generadas con programación genética, estos atributos se describen en las expresiones 28, 29 y 30.

$$x = \cos(x_5 - x_{19}) \tag{28}$$

donde x_5, es *spectral entropy* (señal de audio); x_{19}, es *MFCC* 12 (señal de audio).

$$y = \sqrt{x_{56}} + \frac{2x_{69}}{x_{76} - x_{40}} + x_{69} \tag{29}$$

donde x_{56}, es *delta chroma* 2 (señal de audio); x_{76}, es *HR power* (señal de ritmo cardiaco); x_{40}, es *delta spectral flux* (señal de audio); x_{69} es Valencia (señal de imagen).

$$z = \frac{\sin(x_1) + (\cos(x_{70}) - \sin(x_1) - x_{58})}{\cos(\cos(\tan(x_{58})))} \tag{30}$$

donde x_1, es *energy* (señal de audio); x_{70} es Activación (señal de imagen); x_{58} es *delta chroma* 4 (señal de audio).

En la Figura 21, se muestra el agrupamiento realizado por el algoritmo HPFCM, se destaca la clasificación y distribución de los datos con respecto a cada emoción.

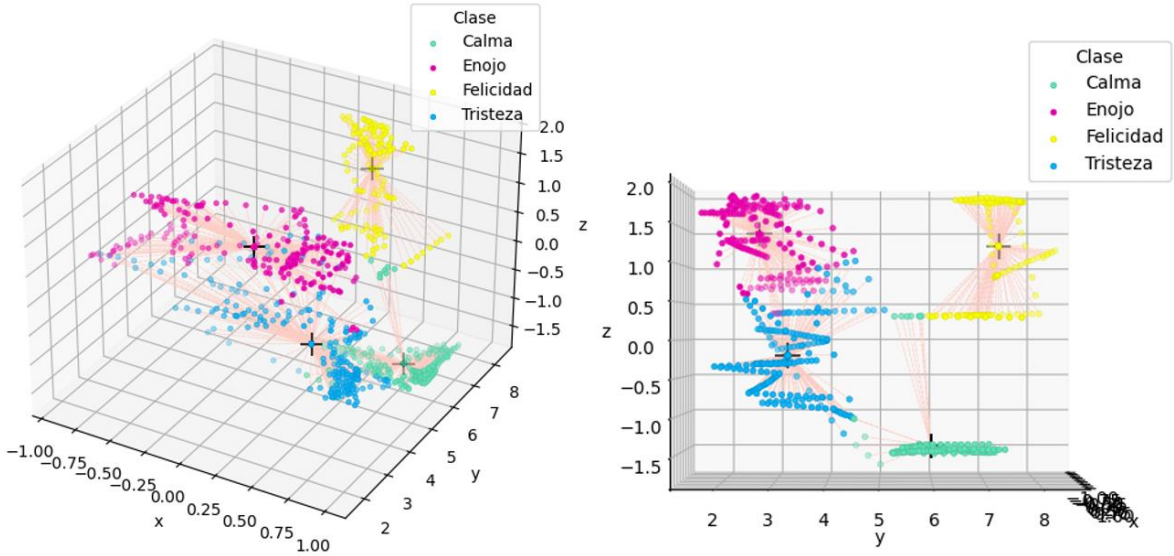


Figura 21. Distribución de datos agrupados con HPFCM.

En la Figura 21, se muestra la distribución de los grupos formados por el algoritmo HPFCM. Una principal observación es el grupo de la clase calma, la mayoría de los objetos se encuentran compactos en su grupo, y muy pocos cerca del grupo Felicidad. Otra observación es la separación de los grupos Enojo, y Tristeza, se visualiza una cercanía entre ambos grupos, causando que algunos objetos cambien de grupo. Sin embargo, para evaluar

estos resultados se emplearon las métricas de evaluación *Accuracy*, *Precision*, *Recall* y *F*1-*Score*. Estas técnicas se muestran en las expresiones 31, 32, 33 y 34.

$$Accuracy = \frac{(VP + VN)}{(VP + FP + FN + VN)} \tag{31}$$

$$Precision = \frac{(VP)}{(VP + FP)} \tag{32}$$

$$Recall \ = \frac{(VP)}{(VP + FN)} \tag{33}$$

$$F1 - Score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \tag{34}$$

La expresión 22, muestra el porcentaje de etiquetas asignadas correctamente por el algoritmo HPFCM; la expresión 23, muestra el nivel de confianza de etiquetas positivas asignadas; expresión 24, muestra el porcentaje de etiquetas positivas asignadas correctamente; la expresión 25, muestra un balance entre los resultados de la expresión 23 y 24.

Para evaluar el desempeño del algoritmo HPFCM en la clasificación de emociones, se construyó una matriz de confusión que compara las etiquetas asignadas por el algoritmo HPFCM con las etiquetas reales del conjunto de datos. La matriz de confusión resultante se muestra en la Figura 22, se puede observar una alta precisión en la clasificación de las cuatro emociones: calma, enojo, felicidad y tristeza.

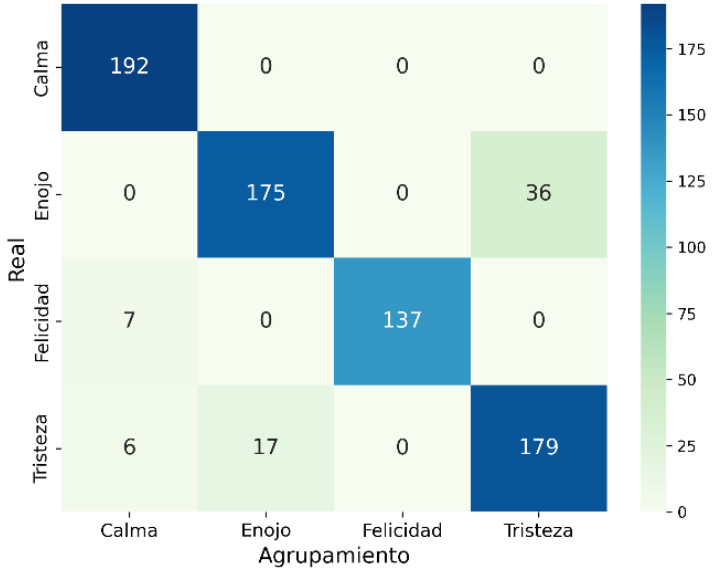


Figura 22. Matriz de confusión.

Como se observa en la Figura 22, la emoción de felicidad fue correctamente clasificado en un 94% de los casos, mientras que enojo y calma alcanzaron un 92% y 90% de aciertos, respectivamente. La emoción tristeza presentó un leve solapamiento con calma, lo que puede atribuirse a la similitud en los patrones de fisiológicos entre estos estados. En la Tabla 21, se muestra el resultado las métricas de evaluación.

Tabla 21. Resultado con métricas de evaluación.

Métrica de evaluación	Resultado
Accuracy	0.9119
Precision	0.9136
Recall	0.9119
*F*1-*Score*	0.9116

Como se observa en la Tabla 21, se obtuvieron buenos resultados con respecto a las métricas de evaluación. Obteniendo valores mayores al 90% en todos los casos, reflejando un desempeño consistente y equilibrado en la clasificación de emociones. En la Tabla 22, se muestran resultados detallados para cada clase.

Tabla 22. Resultados detallados por clase.

Emoción	Precision	Recall	F1-Score
Calma	0.9366	1	0.9673
Enojo	0.9115	0.8294	0.8685
Felicidad	1	0.9514	0.9751
Tristeza	0.8326	0.8861	0.8585

Como se observa en la Tabla 22, las emociones de Felicidad y Calma fueron mejor clasificadas, con valores mayores al 96% de *F*1-*Score*. En contraste, las emociones de Enojo y Tristeza presentaron mayores dificultades, con valores en *Precision* de 91.15% y 83.26%, respectivamente. Este comportamiento puede atribuirse a la superposición de características fisiológicas entre emociones negativas (ver Figura 21).

5.2.6 Análisis de resultados

Con la heurística propuesta HPFCM, se logró reducir la complejidad del algoritmo Fuzzy *C*-Means y mejorar la calidad de solución en la mayoría de las ejecuciones. Con base en los resultados, se observó que de manera general el algoritmo HPFCM mejoró la calidad de la solución y requirió menos iteraciones que el algoritmo Fuzzy *C*-Means. Son destacables los resultados con los conjuntos de datos *WDBC*, *Spam* y *Urban* donde se obtuvieron los mejores

resultados promedio y con *Abalone* los peores. Al resolver el conjunto de datos *SPAM*, con HPFCM y con un valor del $\varepsilon = 0.06$ como umbral de paro, se observó una mejora promedio de la calidad de la solución de un 39.73% y una reducción promedio en el número de iteraciones de 97.65%, con respecto a Fuzzy *C*-Means. Sin embargo, al resolver el conjunto de datos *Abalone* con HPFCM y con un valor de $\varepsilon = 0.9$ como umbral de paro, se observó que el algoritmo HPFCM, obtuvo en promedio una pérdida de calidad del 5.67%, lo cual se considera poco significativo. En contraste, logró una reducción en el número de iteraciones del 98.17%.

Es conveniente mencionar que las ejecuciones de escalabilidad con diferente cantidad de grupos, muestran que la heurística propuesta mantiene una complejidad menor en comparación con el algoritmo Fuzzy *C*-Means y otros algoritmos del estado del arte. Esto permite que su aplicación a cualquier conjunto de datos, pueda mantener una complejidad temporal menor que otros algoritmos de agrupamiento. Reduciendo el tiempo y manteniendo una buena calidad en el agrupamiento.

Los resultados obtenidos con el algoritmo HPFCM usando el conjunto de datos de emociones fueron comparados con los resultados descrito en la investigación de (Colunga, A. *et al*., 2025). En su investigación los autores implementaron una técnica de aprendizaje supervisado (Redes Neuronales Artificiales) para la clasificación de emociones. En sus resultados reportan una *Precision* del 95% con datos de entrenamiento y 94% de Precisión con datos de prueba. En comparación con los resultados del algoritmo HPFCM, se logró una *Precision* global del 91%, lo que muestra un buen desempeño como técnica de aprendizaje no supervisado.

Adicionalmente, los autores destacan la dificultad de clasificar los datos de enojo y tristeza. Por ello, sugieren la integración de otros atributos para diferenciar las clases. Es destacable que, al igual que los resultados que se obtuvieron con la aplicación del algoritmo HPFCM, las emociones de enojo y tristeza presentaron valores de F1-Score de 86.85% y 85.85% respectivamente. Lo que confirma una dificultan de clasificar ambas emociones con técnicas de aprendizaje supervisado y no supervisado.

Capítulo 6

6. Conclusiones y trabajos futuros

En este capítulo, se presentan las conclusiones derivadas del análisis de resultados con las heurísticas desarrolladas en el presente proyecto de investigación. Asimismo, se exponen algunas propuestas con el fin de continuar ampliando la investigación.

6.1 Conclusiones

Las heurísticas P-FCM y HPFCM desarrolladas permiten reducir la complejidad del algoritmo Fuzzy C-Means y por lo tanto se cumplió con el objetivo general. Se observó pérdidas de calidad poco significativas, y ganancias en la calidad de solución cuando se resuelven conjuntos de datos gran tamaño.

1. La heurística P-FCM se basa en el principio de Pareto y contribuye principalmente a la fase de convergencia del algoritmo Fuzzy C-Means. La heurística reduce la cantidad de iteraciones, basándose en el porcentaje acumulado de reducción del valor de la función objetivo por iteración. Con ello se logra reducir la complejidad de Fuzzy C-Means. Al comparar los resultados de P-FCM con Fuzzy C-Means se observaron pérdidas de calidad poco significativas y una reducción considerable de las iteraciones. En el mejor de los casos se redujo la cantidad de iteraciones en un 93.84% usando un valor de umbral de 0.9. En todos los casos, se obtuvo un porcentaje acumulado de reducción del valor de la función objetivo mayor al 90%, en el mejor de los casos fue de 99.9%.
2. HPFCM es una heurística hibrida, que integra aportes a la fase de inicialización y convergencia del algoritmo Fuzzy C-Means. La heurística reduce la cantidad de iteraciones con la inicialización de parámetros optimizados y la evaluación del

porcentaje acumulado de reducción del valor de la función objetivo por iteración. Con ello se logra reducir la complejidad de Fuzzy C-Means y mejorar la calidad de solución. Al comparar los resultados de HPFCM con Fuzzy C-Means se logró reducir la cantidad de iteraciones hasta un 98.17% y una mejora en la calidad de solución del 39.73%.

6.2 Trabajos futuros

Con base a los resultados obtenidos con la presente investigación, se exploraron diferentes áreas de oportunidad para dar continuación con la tesis. A continuación, se proponen trabajos futuros, las cuales abarcan aplicación de las heurísticas en otros entornos de agrupamiento, mejoras en otras fases del algoritmo y aplicaciones en entornos de programación paralela:

a) Extender el uso de las heurísticas desarrolladas a otros algoritmos de agrupamiento como los de partición dura o jerárquicos. Los principios de optimización en la fase de convergencia y la inicialización podrían mejorar la eficiencia computacional, y con ello validar la generalidad de las heurísticas.
b) Integrar el uso de la programación genética cartesiana en la heurística HPFCM con el objetivo de optimizar el cálculo de la métrica de distancia utilizada en la fase de clasificación. Esta integración podría optimizar el cálculo de distancia creando métricas adaptadas a las características de diferentes conjuntos de datos, permitiendo la reducción de la complejidad computacional.
c) Realizar una implementación hibrida de la heurística HPFCM con Redes Neuronales Artificiales para la clasificación de datos. En este enfoque, HPFCM se utilizaría como una etapa de preprocesamiento no supervisado generando las etiquetas suaves, las cuales serviría como parámetros de entrada para las Redes Neuronales Artificiales.
d) Desarrollar un análisis comparativo de la implementación de la heurística HPFCM en entornos paralelos, específicamente en arquitecturas CUDA (GPU) y OpenMP (CPU multicore). En este estudio se evaluarían métricas como la eficiencia paralela y *SpeedUp*, esperando que el uso de estas arquitecturas reduzca los tiempos de ejecución que los entornos secuenciales.

6.3 Logros obtenidos

A continuación, se presentan los principales logros derivados de la presente investigación.

- Pérez, J., Moreno, C., Roblero, S., Almanza, N., Frausto, J., Pazos, R. y Rodríguez, J. (2024). A New Criterion for Improving Convergence of Fuzzy C-Means Clustering, *Axioms*, 13(1), 35. https://doi.org/10.3390/axioms13010035
- Pérez, J., Moreno, C., Roblero, S., Almanza, N., Frausto, J., Pazos, R. y Martínez, A. (2024). Hybrid Fuzzy C-Means Clustering Algorithm, Improving Solution Quality and Reducing Computational Complexity, *Axioms*, 13(9), 592. https://doi.org/10.3390/axioms13090592

Durante el desarrollo de esta investigación, se tuvo oportunidad de colaborar en las siguientes publicaciones:

- Almanza, N., Moreno, C., Roblero, S., Pazos, R., Pérez, O., Landero, V. y Castellanos, V. (2026). Application of Machine Learning to Cluster Analysis of Diabetes Mortality at Municipality Level in Mexico According to Sociodemographic Factors, *Mathematics*, 14(3), 573. https://doi.org/10.3390/math14030573
- Santana, Y., Martínez, A., Clemente, E., Moreno, C. y Mújica, D. (2025). Clustering algorithms: taxonomies, comparative studies, and emerging approaches based on meta-features of datasets, *RA Journal of Applied Research*, 11(11), 1080-1090. https://doi.org/10.47191/rajar/v11i11.16
- Pérez, J., Velasco, E., Almanza, N., Moreno, C. y Martínez, A. (2024). Application of Data Science for the Temporal Analysis of COVID-19 Mortality Data in Mexico, *Tecnología y Ciencia Aplicada*, 7(1), 180-184.

Referencias

Ali, N., El, A. y Cherradi, B. (2021). The performances of iterative type-2 fuzzy C-mean on GPU for image segmentation. *The Journal Supercomputing*, 78, 1583-1601. https://doi.org/10.1007/s11227-021-03928-9

Antoine, V., Guerrero, J. y Romero, G. (2022). Possibilistic fuzzy c-means with partial supervision. *Fuzzy Sets and Systems*, 449, 162-186. https://doi.org/10.1016/j.fss.2022.08.003

Arthur, D. y Vassilvitskii, S. (7-9 de enero de 2007). *k-means++: The Advantages of Careful Seeding*. SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, New Orleans, L.A., U.S.

Bezdek, J.C. (1973). *Fuzzy mathematics in pattern classification*. Cornell Univ. Ithaca, New York, NY, USA.

Bezdek, J. C. (1981). Objective Function Clustering. In M. Nadler, *Pattern Recognition with Fuzzy Objective Function Algorithms* (43-93). Springer. https://doi.org/10.1007/978-1-4757-0450-1_3

Bezdek, J., Ehrlich, R. y Full, W. (1984). FCM: The Fuzzy c-Means clustering algorithm. *Computers & Geosciences*, 10(2-3), 191-203. https://doi.org/10.1016/0098-3004(84)90020-7

Borlea, I. Precup, R., Borlea, A. y Itercan, D. (2021). A Unified Form of Fuzzy C-Means and K-Means algorithms and its Partitional Implementation. *Knowledge-Based Systems*, 214, 106731. https://doi.org/10.1016/j.knosys.2020.106731

Cebeci, Z. (28-30 de septiembre de 2018). *Initialization of Membership Degree Matrix for Fast Convergence of Fuzzy C-Means Clustering*. International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turquía. https://doi.org/10.1109/IDAP.2018.8620920

Cebeci, Z. y Cebeci, C. (2020). A Fast Algorithm to Initialize Cluster Centroids in Fuzzy Clustering Applications. *Information*, 11(9), 1-15. https://doi.org/10.3390/info11090446

Chen, Y., Zhou, S., Zhang, X., Li, D. y Fu, C. (2022). Improved fuzzy c-means clustering by varying the fuzziness parameter. *Pattern Recognition Letters*, 157, 60-66. https://doi.org/10.1016/j.patrec.2022.03.017

Colunga, A., Martínez, A., Estrada, H. y Clemente, E. (2025). Modelo híbrido de programación genética y redes neuronales para el reconocimiento de emociones. *Comp & Syst*, 2025; 29(3), 1457-1470.

Cover, T. and Hart, P. (1967) Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, 13, 21-27. https://doi.org/10.1109/TIT.1967.1053964

Dunn, J. (1974). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3(3), 32–57. https://doi.org/10.1080/01969727308546046

Ezugwu, A., Ikotun, A., Oyelade, O., Abualigah, L., Agushaka, J., Eke, C. y Akinyelu, A. (2022). A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, 110, 104743. https://doi.org/10.1016/j.engappai.2022.104743

Garey, M. y Johnson, D. (1979). NP-Hardness. En W. H. Freeman & Co., *Computers and Intractability: A Guide to the Theory of NP-Completeness* (pp. 109-120).

Ghosh, S. y Kumar, S. (2013). Comparative Analysis of K-Means and Fuzzy C-Means Algorithms. *International Journal of Advanced Computer Science and Applications(IJACSA)*, 4(4), 35-39. https://doi.org/10.14569/IJACSA.2013.040406

Hall, L. y Goldgof, D. (2011). Convergence of the Single-Pass and Online Fuzzy C-Means Algorithms. *IEEE Transactions on Fuzzy Systems*, 19(4), 792-794. https://doi.org/10.1109/TFUZZ.2011.2143418

Hanane, B. y Abdeljabbar, C. (2-4 de diciembre de 2015). *Fast Robust Fuzzy Clustering Algorithm for Grayscale Image Segmentation*. Xème Conférence Internationale: Conception et Production Intégrées, Tánger, Marruecos.

Khang, T., Tran, M. y Fowler, M. (2021). A Novel Semi-Supervised Fuzzy C-Means Clustering Algorithm Using Multiple Fuzzification Coefficients. *algorithms*, 14(9), 258. https://doi.org/10.3390/a14090258

Klir, G. y Yuan, B. (1995). Fuzzy Sets: Basic types. En *Fuzzy Sets and Fuzzy Logic: Theory and Applications* (11-19), Pearson College Div.

Kwok, T., Smith, K., Lozano, S. y Taniar, D. (27-30 de agosto de 2002). *Parallel Fuzzy c-Means Clustering for Large Data Sets*. Euro-Par 2002 Parallel Processing, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45706-2_48

Lan, R. y Fan, J. (14-16 de agosto de 2009). *A Fuzzy C-means Type Clustering Algorithm on Triangular Fuzzy Numbers*. Sixth International Conference on Fuzzy Systems and Knowledge Discovery, Tianjin, China. https://doi.org/10.1109/FSKD.2009.554

Lee, G. y Gao, X. (2021). A Hybrid Approach Combining Fuzzy c-Means-Based Genetic Algorithm and Machine Learning for Predicting Job Cycle Times for Semiconductor Manufacturing. *Applied Science*, 11(16), 7428. https://doi.org/10.3390/app11167428

Lin, Z., Chung, F. y Wang, S. (2009). Generalized Fuzzy C-Means Clustering Algorithm With Improved Fuzzy Partitions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(3), 578-591. https://doi.org/10.1109/TSMCB.2008.2004818

Liu, B., He, S., He, D., Zhang, Y. y Guizani, M. (2019). A Spark-based Parallel Fuzzy *c*-Means Segmentation Algorithm for Agricultural Image Big Data. *IEEE Access*, 7, 42169-42180. https://doi.org/10.1109/ACCESS.2019.2907573

Liu, Q., Liu, J., Li, M. y Zhou, Y. (18-20 de octubre de 2020). *A Novel Initialization Algorithm for Fuzzy C-means Problem*. Theory and Applications of Models of Computation (TAMC 2020), Changsha, China. https://doi.org/10.1007/978-3-030-59267-7_19

Liu, Q., Liu, J., Li, M. y Zhou, Y. (2021). Approximation algorithms for fuzzy C-means problem based on seeding method. *Theoretical Computer Science*, 885(1), 146-158. https://doi.org/10.1016/j.tcs.2021.06.035

Liu, X., Fan, J. y Chen, Z. (2019). Improved fuzzy C-means algorithm based on density peak. *International Journal of Machine Learning and Cybernetics*, 11, 545-552. https://doi.org/10.1007/s13042-019-00993-8

MacQueen, J. (1 January 1967). *Some methods for classification and analysis of multivariate observations*. Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics, Berkeley, California, US.

Mahdi, H., Oskouei, A. y Farajzadeh, N. (2019). New fuzzy C-means clustering method based on feature-weight and cluster-weight learning. *Applied Soft Computing*, 78, 324-345. https://doi.org/10.1016/j.asoc.2019.02.038

Mahid, M., Hosny, K. y Elhenawy, I. (2021). Scalable Clustering Algorithms for Big Data: A Review. *IEEE Access*, 9, 80015-80027. https://doi.org/10.1109/ACCESS.2021.3084057

Manacero, A., Guariglia, E., Souza, T., Lobato, R. y Spolon, R. (2022). Parrallel fuzzy minimals on GPU. *applied sciences*, vol. 12(5), 2385. https://doi.org/10.3390/app12052385

Markelle, K., Longjohn, R. y Nottingham, K., The UCI Machine Learning Repository, Recuperado el 22 Octubre 2023, Disponible en: https://archive.ics.uci.edu.

McCulloch, W.S., Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5, 115–133. https://doi.org/10.1007/BF02478259

Mohammad, S., Kahani, M. y Paydar, S. (2021). Big data fuzzy C-means algorithm based on bee colony optimization using an Apache Hbase. *Journal Big Data*, 8, 64. https://doi.org/10.1186/s40537-021-00450-w

Mújica, D., Gallegos, F. y Rosales, A. (2013). A fuzzy clustering algorithm with spatial robust estimation constraint for noisy color image segmentation. *Pattern Recognition Letters*, 34(4), 400-413. https://doi.org/10.1016/j.patrec.2012.10.004

Mújica, D. y Rubio, J. (2021). Accelerated intuitionistic fuzzy clustering for image segmentation. *Signal, Image and Video Processing*, 15, 1845–1852. https://doi.org/10.1007/s11760-021-01934-1

Nayak, J., Naik, B. y Behera, H. (20-21 de diciembre de 2014). *Fuzzy C-Means (FCM) Clustering Algorithm: A Decade Review from 2000 to 2014*. Computational Intelligence in Data Mining - Volume 2, New Delhi, India. https://doi.org/10.1007/978-81-322-2208-8_14

Parker, J. y Hall, L. (2014). Accelerating Fuzzy-C Means Using an Estimated Subsample Size. *IEEE Transactions on Fuzzy Systems*, 22(5), 1229-1244. https://doi.org/10.1109/TFUZZ.2013.2286993

Pérez, J., Almanza, N. y Romero, D. (2018). Balancing effort and benefit of K-means clustering algorithms in Big Data realms. *PLOS ONE*, 13(9), e0201874. https://doi.org/10.1371/journal.pone.0201874

Pérez, J., Roblero, S., Almanza, N., Frausto, J., Zavala, C., Hernández, Y. Landero, V. (2022). Hybrid Fuzzy C-Means Clustering Algorithm oriented to Big Data Realms. *Axioms*, 11(8), 377. https://doi.org/10.3390/axioms11080377

Pérez, J., Rey, C., Roblero, S., Almanza, N., Zavala, C., García, S. y Landero, V. (2023). POFCM: A Parallel Fuzzy Clustering Algorithm for Large Datasets. *mathematics*, 11(8), 1920. https://doi.org/10.3390/math11081920

Pérez, J., Moreno, C., Roblero, S., Almanza, N., Frausto, J., Pazos, R. y Rodríguez, J. (2024). A New Criterion for Improving Convergence of Fuzzy C-Means Clustering, *Axioms*, 13(1), 35. https://doi.org/10.3390/axioms13010035

Rahimi, S., Zargham, M., Thakre, A. y Chhillar, D. (27-30 de junio de 2004). A Parallel Fuzzy C-Mean Algorithm for Image Segmentation. IEEE Annual Meeting of the Fuzzy Information, 2004. Processing NAFIPS '04, Banff, AB, Canada. https://doi.org/10.1109/NAFIPS.2004.1336283

Ravi, A., Suvarna, A., Souza, A., Mohana, G. y Megha (2012). *A Parallel Fuzzy C Means Algorithm for Brain Tumor Segmentation on Multiple MRI Images*. Proceedings of International Conference on Advances in Computing, Nueva Delhi, India. https://doi.org/10.1007/978-81-322-0740-5_93

Rey, C. (2023) Mejora del algoritmo Fuzzy C-Means mediante el paradigma de programación distribuida y/o paralela [M.S. thesis]. Centro Nacional de Investigación y Desarrollo Tecnológico.

Roblero, S. (2023) *Mejora del algoritmo de agrupamiento Fuzzy C-Means* [Ph.D.]. Centro Nacional de Investigación y Desarrollo Tecnológico.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408. https://psycnet.apa.org/doi/10.1037/h0042519

Ruspini, E. (1969). A new approach to clustering. *Information and Control*, 15(1), 22–32. https://doi.org/10.1016/S0019-9958(69)90591-9

Segovia, N y Guzmán, A. (2025). Evaluación de un chatbot basado en aprendizaje supervisado: impacto en la satisfacción y propuestas de mejora. *Edutec Revista eelectrónica de tecnología educativa*, 92(2025), 234-252.

Shen, Y., Pedrycz, W., Chen, Y., Wang, X. y Gacek, A. (2020). Hyperplane division in Fuzzy C-Means: Clustering Big Data. *IEEE Transactions on Fuzzy Systems*, 28(11), 3032-3046. https://doi.org/10.1109/TFUZZ.2019.2947231

Siringoringo, R. y Jamaluddin, J. (23-24 de noviembre de 2018). *Initializing the Fuzzy C-Means Cluster Center With Particle Swarm Optimization for Sentiment Clustering*. Journal Physics: Conference Series, vol. 1361, Medan, Indonesia. DOI 10.1088/1742-6596/1361/1/012002

Stetco, A., Zeng, X. y Keane, J. (2015). Fuzzy C-means++: Fuzzy C-means with effective seeding initialization. *Expert Systems with Applications*, 42(21), 7541-7548. https://doi.org/10.1016/j.eswa.2015.05.014

Vapnik, V. N. (1995). Methods of Pattern Recognition. In *The nature of statistical learning theory* (pp. 123-167). Springer. https://doi.org/10.1007/978-1-4757-3264-1_6

Vela, V., Mújica, D. y Rubio, J. (2021). Parallel hesitant fuzzy C-means algorithm to image segmentation. *Signal, Image and Video Processing*, 16, 73-81. https://doi.org/10.1007/s11760-021-01957-8

Vela, V. (2022) Agrupamiento difuso titubeante para tareas de segmentación de imágenes y reconocimiento de patrones [Ph.D.]. Centro Nacional de Investigación y Desarrollo Tecnológico.

Verma, H., Verma, D. y Tiwari, P. (2021). A population based hybrid FCM-PSO algorithm for clustering analysis and segmentation of brain image. *Expert Systems with Applications*, vol. 167, 114121. https://doi.org/10.1016/j.eswa.2020.114121

Vimala, S. y Vivekanandan, K. (2019). Kullback–Leibler divergence-based Fuzzy C-Means clustering for enhancing the potential of an movie recommendation system. *SN Applied Science*, 1, 698. https://doi.org/10.1007/s42452-019-0708-9

Wang, X., Wang, Y. y Wang, L. (2004). Improving fuzzy c-means clustering based on feature-weight learning, *Pattern Recognition Letters*, 25(10), 1123-1132. https://doi.org/10.1016/j.patrec.2004.03.008

Weiling, C., Chen, S. y Zhang, D. (2007). Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation. *Pattern Recognition*, 40(3), 825-838. https://doi.org/10.1016/j.patcog.2006.07.011

Wu, Z., Chen, G. y Yao, J. (10-12 de mayo de 2019). *The Stock Classification Based on Entropy Weight Method and Improved Fuzzy C-means Algorithm*. ICBDC '19: Proceedings of the 4th International Conference on Big Data and Computing, Guangzhou, China. https://doi.org/10.1145/3335484.3335503

Xu, R. y Wunsch, D. (2005). Survey of Clustering Algoritms. *IEEE Transactions on Neural Networks*, 16(3), 645-678. https://doi.org/10.1109/TNN.2005.845141

Yang, M. S. (1993). A survey of fuzzy clustering. *Mathematical and Computer Modelling*. 18(11), 1-16. https://doi.org/10.1016/0895-7177(93)90202-A

Yi, C., Tuo, S., Tu, S. y Zhang, W. (2021). Improved fuzzy C-means clustering algorithm based on t-SNE for terahertz spectral recognition. *Infrared Physics & Technology*, 117, 103856. https://doi.org/10.1016/j.infrared.2021.103856

Yu, Q. y Ding, Z. (14-16 de octubre de 2015). *An Improved Fuzzy C-Means Algorithm Based on MapReduce*. 8th International Conference on Biomedical Engineering and Informatics (BMEI), Shenyang, China. https://doi.org/10.1109/BMEI.2015.7401581

Yuan, B., Klir, G. y Swan, J. (20-24 de agosto de 2002). *Evolutionary fuzzy c-means clustering algorithm*. IEEE International Conference on Fuzzy Systems, Yokohama, Japón. https://doi.org/10.1109/FUZZY.1995.409988

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3), 338–353. https://doi.org/10.1016/S0019-9958(65)90241-X

Zhang, Q., Chen, Z. y Leng, Y. (2015). Distributed fuzzy c-means algorithms for big sensor data based on cloud computing. *International Journal of Sensor Networks*, 18(1/2), 32-39.

Zhang, X., Wang, H., Zhang, Y., Gao, X., Wang, G. y Zhang, C. (2021). Improved fuzzy clustering for image segmentation based on a low-rank prior. *Computational Visual Media*, 7(4), 513-528. https://doi.org/10.1007/s41095-021-0239-3