EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación y Desarrollo Tecnológico

Tesis de Maestría

Mejora de un algoritmo de agrupamiento del estado del arte mediante heurísticas aplicadas a la mejora del algoritmo K-Means

presentada por

Lic. Celia Ramos Palencia

como requisito para la obtención del grado de

Maestra en Ciencias de la computación

Director de tesis

Dr. Joaquín Pérez Ortega

Cuernavaca, Morelos, México. Noviembre de 2019.

cenidet
Centro Nacional de Investigación y Desarrollo Tecnológico

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLOGICO NACIONAL DE MÉXICO

Centro Nacional de Investigación y Desarrollo Tecnológico
Departamento de Ciencias Computacionales

"2019, Año del Caudillo del Sur, Emiliano Zapata"

Cuernavaca, Mor., 14/agosto/2019
Oficio No. DCC/079/2019
Asunto: Aceptación de documento de tesis

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial de la **Ing. Celia Ramos Palencia**, con número de control M17CE041, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis profesional titulado **"Mejora de un algoritmo de agrupamiento del estado del arte mediante heurísticas aplicadas a la mejora del algoritmo K-means"** y hemos encontrado que se han realizado todas las correcciones y observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

DIRECTOR DE TESIS

Dr. Joaquín Pérez Ortega
Doctor en Ciencias Computacionales
4795984

REVISOR 1

Dr. José Crispín Zavala Díaz
Doctor en Ciencias Computacionales
3406871

REVISOR 2

Dr. José María Rodríguez Lelis
Doctorado en Ciencias en Ingeniería Mecánica
4500026

C.p. M.E. Guadalupe Garrido Rivera - Jefa del Departamento de Servicios Escolares.
Estudiante
Expediente

NACS/lmz

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLOGICO NACIONAL DE MEXICO

Centro Nacional de Investigación y Desarrollo Tecnológico
Subdirección Académica

"2019, Año del Caudillo del Sur, Emiliano Zapata"

Cuernavaca, Mor., 07/noviembre/2019
No. de Oficio: SAC/294/2019
Asunto: Autorización de impresión de Tesis

LIC. CELIA RAMOS PALENCIA
CANDIDATA AL GRADO DE MAESTRA 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 **"Mejora de un algoritmo de agrupamiento del estado del arte mediante heurísticas aplicadas a la mejora del algoritmo K-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"

DR. GERARDO VICENTE GUERRERO RAMÍREZ
SUBDIRECTOR ACADÉMICO

SEP TecNM
CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
SUBDIRECCIÓN ACADÉMICA

C.p. M.E. Guadalupe Garrido Rivera.- Jefa del Departamento de Servicios Escolares.
Expediente

GVGR/ego

Dedicatoria

A mis padres Enrique y Martina, por ser los pilares en cada uno de los aspectos de mi vida. Este triunfo es de los tres, porque todo lo que he logrado y lo que soy hasta hoy, es el resultado de su empeño en hacer de mí una mujer de bien y lista para enfrentarse al mundo.

A mi hermana Julieta por ser mi cómplice y compañera de aventuras, por motivarme y, a veces, empujarme a hacer cosas que me daban miedo, pero eran por mi bien.

A Jonathan, por vivir muy de cerca este proceso y soportar pacientemente todo lo que conllevó.

A los amigos que el CENIDET y la vida me regalaron: Daniel, Óscar y Luis; porque cada logro, alegría y desesperación no hubiera sido igual sin el toque único de cada uno…Alberto llegaste casi al final, pero lo hiciste para quedarte. Termina esta etapa, pero seguimos juntos.

A Luz, Gudiel y Carlos, gracias por los buenos momentos y las risas, trabajar juntos hizo más ameno todo este tiempo.

A toda mi familia y amigos, porque, aunque estaban lejos, siempre que los necesité me hicieron sentir su presencia.

Al último, pero siempre primero, porque sin ti nada, ni nadie, ni nunca…gracias Señor, gracias Dios.

Agradecimientos

Agradezco al Centro Nacional de Investigación y Desarrollo Tecnológico por la oportunidad de desarrollo durante mis estudios de maestría.

De igual modo, agradezco al Consejo Nacional de Ciencia y Tecnología por el apoyo económico, tanto para mis estudios de maestría como para la estancia realizada en el extranjero, que complementó mi investigación.

A todos y cada uno de los miembros del comité tutorial que se me asignó para esta tesis: Dr. Joaquín Pérez Ortega, Dr. José Crispín Zavala Díaz y Dr. José María Rodríguez Lelis, por cada una de sus observaciones, sugerencias y consejos a lo largo de mi estadía en esta institución.

En especial, agradezco al Dr. Joaquín Pérez Ortega, quién además de fungir como mi director de tesis, fue una ayuda inapreciable, gracias por su paciencia, tiempo, enseñanzas y motivación para formarme realmente como investigadora.

A los Doctores Carlos Pibes y Leandro Balby, de la *Universidade Federal de Campina Grande*, UFCG, Brasil, por el apoyo recibido durante mi estancia académica en dicha institución; así como a mis compañeros del laboratorio de minería de datos, gracias por todo.

Al Dr. Dante Mújica Vargas, por su apoyo académico, no sólo como docente sino también como asesor en temas que fueron de gran ayuda durante mi investigación.

A la Dra. Leticia Sánchez Lima, por su apoyo y asesoría en la redacción de este documento, sus consejos y observaciones fueron de gran ayuda, gracias por su tiempo.

Un agradecimiento en especial para todos aquellos que directa o indirectamente participaron en mi formación académica.

Resumen

El presente estudio propone una mejora para un algoritmo de agrupamiento mediante heurísticas aplicadas al algoritmo K-Means. Dicha mejora se aplicó a la fase de convergencia, para reducir el tiempo de ejecución y disminuir el número de iteraciones que necesita el algoritmo para converger. Esta heurística fue seleccionada dentro de todas las que se han realizado en el Centro de Investigación y Desarrollo Tecnológico CENIDET.

Para cumplir el objetivo propuesto, el cual es mejorar un algoritmo de agrupamiento a través de un nuevo criterio de paro, en esta investigación se realizó un estudio entre varios algoritmos de agrupamiento, como, por ejemplo, *Gustafson-Kessel, Mean-Shift y Fuzzy C-Means,* siendo esté el que se seleccionó para ser mejorado mediante la implementación de una heurística orientada a la mejora del algoritmo K-Means. Asimismo, se hizo una revisión de los trabajos que se han hecho en el CENIDET que abordan la temática acerca de las mejoras aplicadas al algoritmo K-Means, por ejemplo, *N-means, Early Stop K-Means, Early Classification, HC Heuristics* y *OK-Means* para que, a partir de su análisis se pudiera proponer una nueva mejora para el algoritmo de agrupamiento seleccionado. Los experimentos fueron llevados a cabo con seis instancias de datos obtenidas del repositorio de la Universidad de California UCI, utilizado ampliamente por la comunidad científica. El modelo utilizado como base para dicha propuesta fue el de la heurística OK-Means, por ser la más reciente y contar con la novedad de ser aplicada en la fase de convergencia de K-Means, a diferencia de las otras, que en su mayoría son orientadas a las fases de inicialización o clasificación.

El resultado final de esta investigación fue la propuesta de una nueva heurística aplicada a la fase de convergencia del algoritmo Fuzzy C-Means, denominada *Optimized Fuzzy C-Means*. Esta heurística establece un nuevo criterio de paro para el algoritmo, disminuyendo el tiempo de ejecución y el número de iteraciones necesarias para detenerse, sin una pérdida considerable de la calidad.

Abstract

This study proposes an improvement for a clustering algorithm using heuristics applied to K-Means algorithm. This improvement was applied to the convergence step, to reduce the execution time and decrease the number of iterations that the algorithm needs to converge. This heuristic was selected among all those that have been made in CENIDET.

To meet to the objective proposed in this investigation, a study was carried out among several clustering algorithms, as, for example, Gustafson-Kessel, Mean-Shift and Fuzzy C-Means, being the one that was selected to be improved by implementing a heuristic aimed at improving the K-Means algorithm. Likewise, a review was made of the work that has been done in CENIDET that addresses the issue of improvements applied to the K-Means algorithm, for example, N-means, Early Stop K-Means, Early Classification, HC Heuristics and OK-Means, so that, from its analysis, a new improvement can be proposed for the selected clustering algorithm. The experiments were carried out with six instances of data obtained from the repository of the University of California UCI, widely used by the scientific community.

The model used as the base for this proposal was that of the OK-Means heuristic, being the most recent and having the novelty of being applied in the convergence step of K-Means, unlike the others, which mostly are oriented to the initialization or classification steps.

The result of this investigation was the proposal of a new heuristic applied to the convergence step of the Fuzzy C-Means algorithm, called *Optimized Fuzzy C-Means*, which establishes a new stop criterion for the algorithm, reducing the execution time and the number of iterations necessary to stop, without a considerable loss of quality.

Tabla de contenido Página

Dedicatoria……..……I

Agradecimientos…….……II

Resumen…….....…...III

Abstract…….……..……..…IV

Lista de Figuras……...…..VI

Lista de Tablas……...VI

Capítulo 1 Introducción 1

1.1. Introducción 2

1.2. Contexto de la investigación en el CENIDET 3

1.3. Descripción del problema de investigación 4

1.4. Justificación 5

1.5. Objetivos 5

1.5.1. Objetivo general 5

1.5.2. Objetivos específicos 5

1.6. Alcances y limitaciones 5

1.6.1. Alcances 5

1.6.2. Limitaciones 6

1.7. Organización del documento 6

Capítulo 2 Revisión del estado del arte 7

2.1. Mejoras al algoritmo K-Means desarrolladas en CENIDET 8

2.2. Marco teórico 11

2.2.1. Algoritmos de agrupamiento 11

2.2.2. Algoritmos de agrupamiento difuso 13

2.2.3. Algoritmo Fuzzy C-Means 14

2.2.4. Pseudocódigo y complejidad de Fuzzy C-Means 17

2.2.5. Mejoras al algoritmo Fuzzy C-Means 19

2.2.6. Heurística O-K-Means 21

Capítulo 3 Integración y adecuación del algoritmo Fuzzy C-Means con la heurística O-K-Means 24

3.1. Proceso de experimentación con Fuzzy C-Means 25

3.2. Resultados de la ejecución del algoritmo Fuzzy C-Means 26

3.3. Correlación entre objetos que cambian de grupo y función objetivo 29

3.4. Propuesta de mejora con la heurística OFCM 32
3.5. Determinando el valor del umbral 33
Capítulo 4 Pruebas para la mejora de OFCM y análisis de resultados 38
4.1. Descripción de los casos de prueba 39
Capítulo 5 Conclusiones y trabajos futuros 43
5.1. Conclusiones 44
5.2. Trabajos futuros 44
REFERENCIAS 45
Anexo A 48

Lista de Figuras

Página

Figura 1. Descripción del problema 4
Figura 2. Agrupamiento de datos 12
Figura 3. Clasificación de los algoritmos de agrupamiento 12
Figura 4. Relación cuasi-lineal para mostrar la correlación entre γ y δ 30
Figura 5. Correlación entre γ y δ 31
Figura 6. Muestra parcial del porcentaje de iteraciones y de la calidad aplicando el diagrama de Pareto 36
Figura 7. Total de iteraciones entre Fuzzy C-Means y OFCM k=3 41
Figura 8. Total de iteraciones entre Fuzzy C-Means y OFCM k=4 41
Figura 9. Total de iteraciones entre Fuzzy C-Means y OFCM k=5 42
Figura A 1. Diferencia de tiempo de ejecución entre Fuzzy C-Means y OFCM…….49
Figura A 2. Total de iteraciones de Fuzzy C-Means y OFCM…….49

Lista de Tablas

Página

Tabla 1. Pseudocódigo Fuzzy C-Means 18
Tabla 2. Pseudocódigo algoritmo O-K-Means 22
Tabla 3. Instancias utilizadas para la fase de experimentación 26
Tabla 4. Resultados de la ejecución de la instancia Ecoli con el algoritmo Fuzzy C-Means 28
Tabla 5. Resultados de la instancia Ecoli con porcentaje de objetos cambiando de grupo y diferencias de la función objetivo 29
Tabla 6. Descripción del pseudocódigo Optimizaded Fuzzy C-Means 33
Tabla 7. Relación entre el esfuerzo computacional y la calidad de solución con la instancia Ecoli 35
Tabla 8. Cambio de objetos por iteración de la instancia Ecoli 37
Tabla 9. Resultados que muestran el valor de la función objetivo, el número de iteraciones y el valor de umbral k=3 39
Tabla 10. Resultados que muestran el valor de la función objetivo, el número de iteraciones y el valor de umbral k=4 40
Tabla 11. Resultados que muestran el valor de la función objetivo, el número de iteraciones y el valor de umbral k=5 40
Tabla A 1. Resultados de reducción de tiempo y diferencia de la calidad de solución con Fuzzy C-Means y OFCM 48

Capítulo 1
Introducción

El que lee mucho y anda mucho, ve mucho y sabe mucho.
Miguel de Cervantes Saavedra

Don Quijote de la Mancha.

1.1. Introducción

Esta investigación se ubica en el contexto general del problema de agrupación de objetos, el cual ha sido estudiado ampliamente a causa de sus aplicaciones a una gran variedad de campos. Estos incluyen ingeniería (aprendizaje de máquina, inteligencia artificial, reconocimiento de patrones, ingeniería mecánica, ingeniería eléctrica, entre otras.); ciencias computacionales (ingeniería de software, minería de datos, análisis de datos espaciales, colección de documentos textuales, segmentación de imágenes); ciencias de la vida y medicina (genética, biología, microbiología, paleontología, psiquiatría, patología); ciencias de la tierra (geografía, geología, sensores remotos); y economía (negocios, comercialización), por mencionar solamente algunas de las aplicaciones.

Se han propuesto diversos algoritmos de agrupamiento que particionan conjuntos de objetos en grupos de acuerdo con la similitud de sus atributos. Uno de los más populares es el algoritmo de agrupamiento K-Means [1] porque su implementación es relativamente simple y sus resultados son fáciles de interpretar. Sin embargo, una de sus principales limitaciones es su alto costo computacional. En consecuencia, su mejora sigue siendo un problema abierto por su relevancia y vigencia.

En investigaciones anteriores y en las que actualmente están en desarrollo en el Centro de Investigación y Desarrollo Tecnológico CENIDET, se ha mejorado la eficiencia y eficacia de K-Means, en diferentes fases del mismo. Tales mejoras han sido objeto de publicaciones en revistas especializadas. En particular, en esta investigación, se desea mejorar un algoritmo de agrupamiento del estado del arte mediante heurísticas aplicadas a la mejora del algoritmo K-Means.

El presente capítulo está organizado de la siguiente manera: en la Sección 1.1 se describe el contexto de la investigación; en la Sección 1.2 se describe el problema a resolver por medio de esta investigación; la Sección 1.3 plantea la hipótesis de la investigación; en la Sección 1.4 se describe la justificación; en la Sección 1.5 se muestran los objetivos de esta tesis; la Sección 1.6 presenta los alcances y limitaciones de la misma, y por último; en la Sección 1.7 explica la organización del presente documento.

1.2. Contexto de la investigación en el CENIDET

En el CENIDET, desde el año 2005, se ha incursionado en la mejora y aplicación del algoritmo K-Means. Algunas de estas investigaciones son las siguientes:

a) Basave Torres, "Desarrollo de un Mecanismo de Agrupamiento Aplicado al Problema de Selección de Algoritmos Heurísticos", Tesis de maestría, (2005).
b) Moreno Hernández, "Mejora del algoritmo K-Means incrementando su eficiencia en la fase de clasificación", Tesis de maestría. (2013).
c) López Vitervo, "Incremento de la eficiencia del algoritmo K-Means mediante la mejora de la heurística Early Classification". Tesis de maestría. (2015).
d) Adams López, "Adecuación de una heurística del algoritmo K-Means para mejorar un algoritmo de particionamiento". Tesis de maestría. (2017).
e) Ortega Almanza, "Desarrollo de heurísticas para la mejora del algoritmo K-Means en las fases de clasificación y convergencia". Tesis de doctorado. (2018).

En las anteriores investigaciones fueron desarrolladas heurísticas con el fin de mejorar el algoritmo K-Means, las cuales han mostrado resultados alentadores. Estas mejoras motivaron el desarrollo de la presente tesis de maestría, con la finalidad de implementar una heurística con un algoritmo distinto de K-Means. Las heurísticas desarrolladas hasta ahora para mejorar el algoritmo, se encuentran orientadas en las fases de inicialización y clasificación [1]. Esta investigación se propone desarrollar una heurística en la fase de convergencia.

1.3. Descripción del problema de investigación

En la Figura 1, que se muestra a continuación, se observa el problema central de esta investigación, el cual consiste en integrar un algoritmo de agrupamiento del estado de arte con una heurística orientada a la mejora del algoritmo K-Means para dar como resultado un algoritmo de agrupamiento mejorado con dicha heurística.

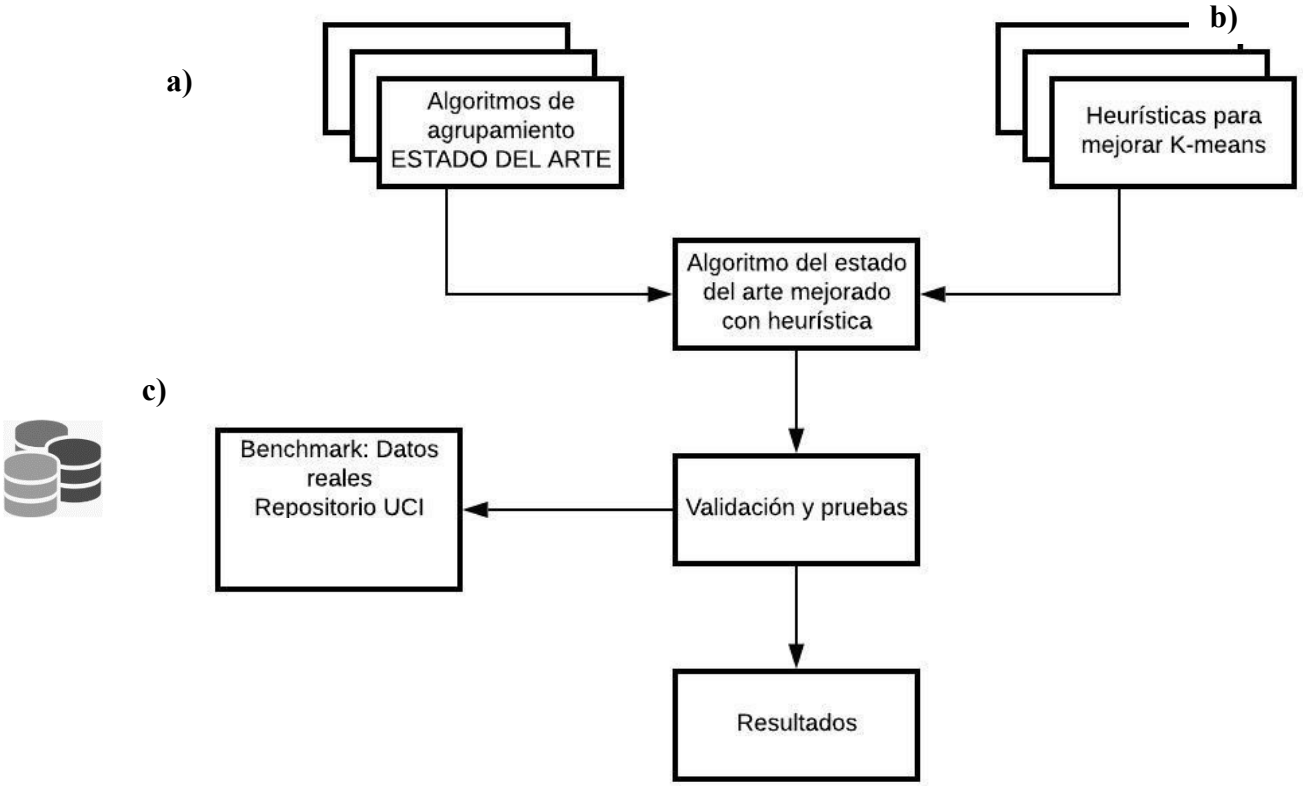

Figura 1. Descripción del problema

Para resolver el problema que motivó esta investigación, hubo que estudiar diversos algoritmos de agrupamiento del estado del arte, por ejemplo: *Mean-Shift, Fuzzy C-Means* y *Gustafson-Kessel* (**a**), Para seleccionar uno de ellos. Asimismo, se estudiaron diversas heurísticas que se han desarrollado en el CENIDET para mejorar K-Means, por ejemplo, *Early Stop Clasification, N-means, Honey Comb*. De igual manera, se seleccionó una de ellas (**b**), para después integrar ambos con el fin de obtener un algoritmo del estado del arte mejorado. Para la realización y validación de pruebas (**c**) se utilizaron datos de instancias obtenidas del repositorio de la Universidad de California, Irvine (UCI).

1.4. Justificación

La complejidad de los algoritmos de agrupamiento sigue siendo un problema actual y vigente gracias a su gran aplicación en diversos campos de estudio. Por ejemplo: minería de datos, reconocimiento de patrones, procesamiento de imágenes, ingeniería de software, medicina, agricultura, entre otros.

En los últimos años se ha incrementado la cantidad de datos generados por aplicaciones computacionales, sensores, redes sociales, y las instancias o *data sets* han crecido en tamaño exponencial. Por lo tanto, se requiere de algoritmos de agrupamiento más eficaces para su solución y análisis. Para resolver estas instancias se requiere de mucho tiempo, por lo cual, es importante desarrollar algoritmos que ayuden a mejorar este aspecto, y a su vez, heurísticas que mejoren la eficiencia de dichos algoritmos.

A causa de lo anterior se propone una heurística orientada al mejoramiento del algoritmo K-Means que pueda aplicarse a un algoritmo de agrupamiento, reduciendo su tiempo de ejecución y el número de iteraciones necesarias para su convergencia, sin disminuir de manera significativa la calidad.

1.5. Objetivos

1.5.1. Objetivo general

Mejorar un algoritmo de agrupamiento del estado del arte mediante la integración de heurísticas orientadas a la mejora del algoritmo K-Means.

1.5.2. Objetivos específicos

1) Obtener una mejora de un algoritmo de agrupamiento del estado del arte.
2) Implementar de manera computacional el algoritmo mejorado.
3) Validar y probar los resultados obtenidos.

1.6. Alcances y limitaciones

1.6.1. Alcances

Al concluir esta investigación, se obtendrá un algoritmo de agrupamiento mejorado.

1.6.2. Limitaciones

1) La implementación de la heurística en el algoritmo será probada con el equipo disponible en CENIDET.
2) La validación será experimental.
3) Los experimentos para la implementación de la mejora serán realizados con instancias reales de repositorios de la UCI.
4) La experimentación con la heurística y el algoritmo seleccionados se hará durante el estudio de los mismos.

1.7. Organización del documento

El presente documento está organizado de la siguiente manera: en el Capítulo 2 se presenta un estudio adecuado del estado del arte acerca del algoritmo de agrupamiento seleccionado, así como el uso de las heurísticas. En el Capítulo 3, se desarrolla la integración y adecuación de la heurística en el algoritmo, con la correspondiente descripción de sus componentes. En el Capítulo 4, se muestra la validación experimental del algoritmo ya mejorado y se presentan los resultados obtenidos con sus respectivos análisis. En el Capítulo 5, se presentan las conclusiones derivadas de la investigación, así como algunas propuestas para trabajos futuros.

Capítulo 2
Revisión del estado del arte

Nada en la vida es para ser temido, es sólo para ser comprendido. Ahora es el momento de entender más, de modo que podamos temer menos.
Marie Curie

Gracias a los avances en el área de las tecnologías de la información y con ellos, el uso creciente de usuarios, día con día se genera una cantidad impresionante de datos, los cuales, es necesario almacenar y administrar de manera factible para realizar en su debido momento, su estudio y análisis de forma acertada, óptima y con menor tiempo. Asimismo, la mejora de algoritmos es un tema actual y vigente por su gran aplicación en diversas áreas del conocimiento.

En este capítulo se describen avances reportados en artículos de investigación los cuales proponen mejoras a algunos algoritmos de agrupamiento a través del uso de una heurística desarrollada para mejorar el algoritmo K-Means. Se realizó un estudio de éstos para poder seleccionar uno e integrarlo con un algoritmo de agrupamiento.

2.1. Mejoras al algoritmo K-Means desarrolladas en CENIDET

En esta sección se describirán las investigaciones más recientes que se desarrollaron con el objetivo de lograr la mejora de algoritmos de agrupamiento mediante heurísticas desarrolladas para mejorar el algoritmo K-Means.

a) En *Early Classification: A New Heuristic to Improve the Classification Step of K-Means* [1], se propuso una nueva heurística para reducir el número de cálculos necesarios en la etapa de clasificación del algoritmo K-Means, sin una pérdida significativa de su calidad. Esto se logró usando información estadística acerca del desplazamiento de los centroides en cada iteración. Esta heurística fue llamada *Early Classification* (EC). La ventaja de EC es que identifica y excluye de futuros cálculos, aquellos objetos que, de acuerdo a un umbral de equidistancia, presentan baja probabilidad de cambiar de grupo en iteraciones subsecuentes. Su objetivo principal, fue reducir el número de cálculos necesarios en la etapa de clasificación del algoritmo K-Means. Se demostró que es posible mejorar este algoritmo durante su etapa de clasificación.
b) En *Improving the Efficiency and Efficacy of the K-means Clustering Algorithm Through a New Convergence Condition* [2], se desarrolló una mejora del algoritmo K-Means estándar usando una nueva condición durante su etapa de convergencia. La condición que se propuso fue incorporar el error cuadrático, lo cual garantiza que el algoritmo se detenga en un óptimo local reduciendo el número de iteraciones y mejorando la calidad de la solución.

   Es importante remarcar que la mejora propuesta no fue compatible con otras técnicas para mejorar K-Means, porque estas técnicas fueron aplicadas a las etapas de inicialización y clasificación del

algoritmo. Sin embargo, fue posible combinarla con otras variantes de K-Means, contribuyendo a mejorar su rendimiento.

c) En *Mejora del algoritmo k-means mediante una meta-heurística orientada a la reducción de su complejidad computacional* [3], se propuso una nueva meta-heurística denominada N-Means, que permitió reducir la complejidad computacional de K-Means de manera importante. Además, posibilitó que con los mismos recursos computacionales se resolvieran instancias más grandes en menor tiempo. Con esta meta-heurística se obtuvieron reducciones de tiempo hasta del 91% y la disminución de la calidad fue de sólo 5.5%. Es destacable que con base en el análisis de los resultados se observó un comportamiento cuasi lineal de N-means. Con esta investigación se demostró que es factible reducir la complejidad del algoritmo K-Means de manera importante, mediante la heurística propuesta.

d) En *Improvement to the K-Means Algorithm Through a Heuristics Based on a Bee Honeycomb Structure* [4], se implementaron nuevas heurísticas para reducir la complejidad computacional en la etapa de clasificación del algoritmo K-Means. Dichas heurísticas están inspiradas en la estructura de un panal de abejas que el algoritmo construye cuando los objetos se visualizan en un espacio bi-dimensional. Esto permitió observar cómo un objeto puede cambiar de pertenencia hacia los grupos vecinos. Las heurísticas mencionadas consisten en realizar cálculos de distancia sólo con respecto a los centroides de los grupos vecinos.

   En esta investigación se demuestra que es posible reducir el tiempo de ejecución del algoritmo K-Means usando heurísticas inspiradas biológicamente. Este conocimiento fue aplicado en la etapa de clasificación del algoritmo. Por lo tanto, la distancia de un objeto se calculó sólo con sus ocho grupos adyacentes, excluyendo de estos cálculos al resto de los centroides. Las heurísticas propuestas en este estudio pueden combinarse con otras técnicas de mejora del tiempo de ejecución del algoritmo K-Means.

e) En *An Improvement to the K-means Algorithm Oriented to Big Data* [5], se aprovechó el problema de la complejidad computacional de K-Means para hacer posible la solución de grandes volúmenes de datos, por ejemplo, *Big Data*, sin que esto implique una pérdida considerable de su calidad. Como resultado se propusieron nuevas meta-heurísticas, las cuales, por una asignación temprana de objetos a grupos, redujeron significativamente el número de cálculos de distancia de objetos a centroides. Gracias al uso creciente de *Big Data,* es importante mejorar la eficacia de los algoritmos

para procesar grandes cantidades de datos con tiempos razonables de ejecución sin pérdida de su calidad.

f) En *Improvement to the K-Means algorithm by using its geometric and cluster neighborhood properties* [6], se aplicó la técnica denominada *vecino más cercano* con el objetivo de disminuir el número de centroides involucrados en los cálculos de distancia del algoritmo. El resultado fue la reducción de la complejidad de K-Means. La principal contribución de este trabajo consistió en realizar el análisis del grupo más cercano y particularmente en relacionar la calidad de solución y el tiempo de ejecución con el número de grupos de vecinos.

g) En *Optimization of the K-Means algorithm for the solution of high dimensional instances* [7], se desarrolló una nueva heurística para reducir la complejidad computacional del algoritmo K-Means. Esta heurística se derivó de la observación visual del proceso de agrupamiento de K-Means, en el cual se determinó que los objetos sólo pueden migrar hacia grupos adyacentes sin cruzar con grupos distantes. Además, esta heurística redujo significativamente el número de cálculos de distancia de un objeto hacia los centroides de los grupos a los que potencialmente puede clasificarse. El objetivo principal de esta heurística fue diferente al reportado en el estado del arte, porque redujo los cálculos de distancia de los objetos a los centroides durante la etapa de clasificación. Para cada objeto, los cálculos se realizaron sólo con relación a los centroides de los grupos adyacentes.

h) En *Improving the Efficiency of the K-medoids clustering Algorithm by Getting Initial Medoids* [8], se aplicó una heurística para mejorar un algoritmo de agrupamiento denominado K-Medoids. Esta heurística fue desarrollada originalmente para K-Means. El algoritmo K-Medoids es uno de los más usados en el área de agrupamiento de datos. Sin embargo, una de sus limitaciones es su sensibilidad a los medoides iniciales. En el artículo, se propone la generación de medoides iniciales óptimos, los cuales fueron obtenidos a través de dos pasos: en el primero, los datos se agruparon con una variante eficiente del algoritmo K-Means denominada *Early Classification*. En el segundo paso, los centroides generados por K-Means fueron transformados en medoides iniciales óptimos. Con este procedimiento, se demostró que es posible mejorar la eficiencia y eficacia de un algoritmo de agrupamiento diferente de K-Means con una mejora que fue desarrollada para dicho algoritmo.

i) En *Desarrollo de heurísticas para la mejora del algoritmo K-Means en las fases de clasificación y convergencia* [9] se propuso un criterio para balancear el tiempo de procesamiento y la calidad de solución de los algoritmos de agrupamiento K-Means cuando se aplicaron en instancias en las

cuales el número *n* de objetos sea grande. La mayoría de las estrategias conocidas para mejorar el rendimiento de los algoritmos K-Means están relacionadas con las etapas de inicialización y clasificación. En esta investigación, el criterio se aplicó para la etapa de convergencia. Con este mismo criterio, en la investigación que se propone desarrollar, se tratará de integrar una heurística desarrollada inicialmente para K-Means con un algoritmo de agrupamiento diferente.

2.2. Marco teórico

En esta sección, se explicarán detalladamente los conceptos que servirán de base para desarrollar y comprender el proceso de la presente investigación.

2.2.1. Algoritmos de agrupamiento

El agrupamiento de datos es uno de los temas de estudio más importantes en el área de aprendizaje no supervisado principalmente a sus posibles aplicaciones prácticas. Puede ser definido como un método que se utiliza para crear un determinado número de grupos de objetos, de tal forma que los objetos pertenecientes a un grupo son muy similares entre sí y, al mismo tiempo, dichos objetos son distintos respecto a los objetos de otros grupos (Figura 2) [10].

La principal tarea del agrupamiento consiste en encontrar una estructura en la colección de datos no etiquetados. El agrupamiento de datos es aplicado a una gran variedad de dominios, tales como: exploración de datos científicos, diagnóstico médico, biología, astronomía, investigación de mercados, procesamiento de texto o imágenes y minería de datos, entre otros. Por ejemplo, en la Figura 2, se observa el agrupamiento de datos de acuerdo a su similitud en tres grupos definidos, así como los centroides de cada uno [10].

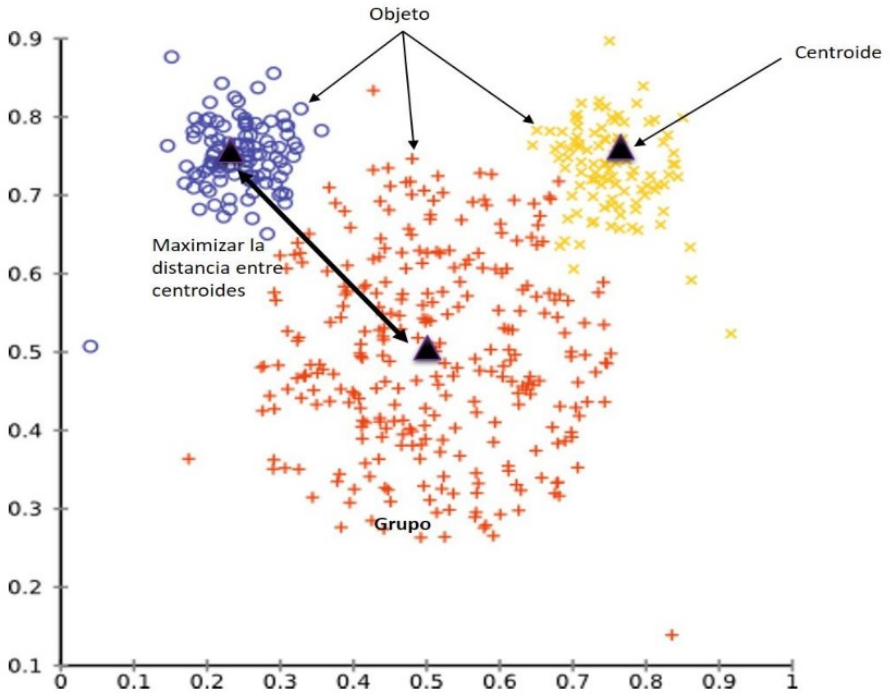

Figura 2. Agrupamiento de datos

En los últimos años, se ha desarrollado una amplia variedad de algoritmos de agrupamiento, los cuales se pueden clasificar en jerárquicos y particionales, tal como se muestra en la Figura 3:

Figura 3. Clasificación de los algoritmos de agrupamiento

Los algoritmos jerárquicos conforman una segmentación de los datos con forma de estructura de árbol. De esta manera se establece la relación entre el nodo padre y los nodos hijos, los cuales se ramifican exponencialmente. Por su parte, los algoritmos particionales dividen ese conjunto de datos en subconjuntos o grupos, en función de la proximidad de dichos objetos a un representante o centroide elegido, por lo general, aleatoriamente para cada grupo.

En la actualidad debido a su creciente demanda, se han desarrollado con mayor frecuencia diversos algoritmos de agrupamiento. Sin embargo, una de las limitantes que presentan es su alto costo computacional. Lo cual será el tema de la presente investigación.

En esta investigación, se decidió delimitar el estudio de los algoritmos de agrupamiento a solamente tres. Una vez revisada la literatura, se optó por seleccionar un solo algoritmo cuyo funcionamiento fuera similar a K-Means. Se decidió aplicar un algoritmo de agrupamiento difuso, el cual, a diferencia de K-Means, que es un algoritmo de partición dura, funciona con partición suave. El algoritmo seleccionado se denomina Fuzzy C-Means. Más adelante se describirán los conceptos anteriores de forma precisa.

2.2.2. Algoritmos de agrupamiento difuso

Dentro de la clasificación de algoritmos de agrupamiento se encuentran los de agrupamiento difuso, los que forman una clase de algoritmos de agrupamiento en la cual cada uno de los elementos tiene un grado de pertenencia a los grupos. Este tipo de algoritmos surge por la necesidad de resolver una deficiencia del agrupamiento exclusivo, el cual considera que cada elemento se puede agrupar inequívocamente con los elementos de su grupo y que, por lo tanto, no se asemeja al resto de los elementos [11].

Tras la introducción de la lógica difusa surgió una solución para ese problema, la cual consiste en representar la similitud entre un elemento y un grupo por una función, llamada función de pertenencia, que toma valores entre cero y uno. Los valores cercanos a uno indican una mayor similitud, mientras que los cercanos a cero indican una menor similitud. Por lo tanto, el problema del agrupamiento difuso se reduce a encontrar una caracterización óptima para encontrar valores cercanos a uno.

En la clasificación que se presenta en la siguiente lista, se observa que el algoritmo Fuzzy C-Means pertenece a la clase de agrupamiento difuso clásico. En [12] se puede encontrar una breve, pero clara descripción de los distintos algoritmos que pertenecen a esta clase:

- Gustafson-Kessel (GK)
- Gath-Geva
- Mean-Shift
- Fuzzy C-Means

En los siguientes párrafos, se describirán brevemente las características de cada uno de los algoritmos mencionados en la lista anterior:

- *Gustafson-Kessel* es uno de los algoritmos de agrupamiento difuso cuya función es medir la pertenencia de un objeto a un centroide en lugar de la tradicional clasificación binaria 0/1 (pertenezca o no pertenezca). De manera que al final, cada objeto no se asigna a un centroide, sino que se le otorga un vector de medidas de membresía. Por lo tanto, un objeto generalmente pertenece a todos los grupos de forma simultánea, pero su pertenencia tiene una "fuerza" diferente para cada uno de los diferentes grupos. [13]
- El algoritmo *Gath-Geva* [14] permite detectar conjuntos de objetos de distintas formas, tamaños y densidades. A causa del tipo de distancia que se utiliza (Euclidiana, Manhattan, Mahalanobis, entre otras), los conjuntos no están restringidos a un solo grupo como en el caso de K-Means. Sin embargo, es muy sensible a la inicialización, ya que el algoritmo tiende a converger en óptimos locales.
- *Mean-Shift* es un algoritmo no paramétrico que puede ser usado tanto para agrupamiento como para segmentación, entre otros usos. Encuentra las modas de distribuciones, pero sin cuantificar cuántas modas existen. Considera que el espacio de datos es una función de densidad de probabilidad muestreada. Para cada punto del conjunto de datos, encuentra la moda más cercana. Para ello, define una región alrededor de ese punto y encuentra su media, cambiando la posición de la media actual a una nueva (*shift*), y repitiendo el proceso hasta converger [15].

Ya que el algoritmo seleccionado para desarrollar su mejora, fue Fuzzy C-Means, se describirá con mayor detalle en el siguiente apartado.

2.2.3. Algoritmo Fuzzy C-Means

Los algoritmos de agrupamiento difuso permiten suprimir el requerimiento de que los objetos tengan que ser asignados a uno y solo uno de los grupos. El algoritmo Fuzzy C-Means es un algoritmo de agrupamiento derivado de K-Means. Ambos algoritmos están basados en funciones objetivo *J*, las cuales son criterios matemáticos que cuantifican la calidad de los modelos en el particionamiento de los datos [11].

Las funciones objetivo sirven como *funciones de costo* que deben ser minimizadas para obtener soluciones de agrupamiento óptimas. La tarea de agrupamiento puede ser formulada como el problema de optimizar una función. Esto es, los algoritmos determinan la mejor descomposición de un conjunto de datos en un número predefinido de clases o grupos minimizando su función objetivo.

En sus formas más básicas ambos algoritmos buscan un número predefinido de *c* grupos en un conjunto de datos, donde cada grupo está representado por su centroide. Sin embargo, difieren en la forma en que se asignan los datos a cada grupo. En un análisis de agrupamiento, en *K-Means* cada dato es asignado a un solo grupo. En Fuzzy C-Means, los datos pueden pertenecer a más de un grupo y también tener diferentes grados de pertenencia en los diferentes grupos [11].

El análisis de agrupamiento difuso permite encontrar la pertenencia gradual de los datos a los centros de los grupos medidos como grados en un intervalo de [0,1]. Esta característica aporta una mayor flexibilidad para expresar que los datos pueden pertenecer a más de uno de los grupos.

Fuzzy C-Means es uno de los algoritmos para agrupamiento de partición difusa más difundido dentro de la comunidad científica. Se utiliza en múltiples ámbitos que van desde las ciencias sociales hasta la ingeniería. Es recomendado para el reconocimiento de patrones, la segmentación de imágenes, agrupamiento de imágenes, *Big Data*, entre otros.

El algoritmo Fuzzy C-Means fue introducido por Dunn [16] y ampliado por Bezdek [17]. Bezdek y Dunn presentaron un método de agrupamiento que combinaba los conceptos de los métodos basados en función objetivo con los de la lógica difusa. De esta manera, un objeto podría tener distintos grados de pertenencia en los diferentes subgrupos resultantes, en lugar de poseer solamente una pertenencia discreta (0 ó 1) [11].

A continuación, se realizará una descripción matemática del algoritmo Fuzzy C-Means.

Sea X un conjunto de *N* objetos,

se dice que una partición $\mathrm{P} = \{c_1,\ c_2, \ldots, c_c\}$,

es una partición difusa de **X** si y solo si se cumple con:

$$\sum_{i=1}^{c} \mu_{ij} = 1, j = 1, 2, \ldots, N \qquad (1)$$

$$1 \geq \mu_{ij} \geq 0, i = 1, 2, \ldots, c, j = 1, 2, \ldots, N \qquad (2)$$

$$n > \sum_{j=1}^{N} \mu_{ij} > 0, i = 1, 2, \ldots, c \qquad (3)$$

Donde, μ_{ij} es el grado de pertenencia del objeto *i* al centroide *j*.

- En (1) la suma de los grados de pertenencia de un objeto *i* a los distintos grupos *j* debe ser igual a 1.
- En (2) el grado de pertenencia de un objeto *i* al centroide *j* debe estar entre 0 y 1.
- En (3) la suma de todos los grados de pertenencia en un grupo tiene que ser mayor a 0 y menor que *N*, es decir, no se pueden tener grupos vacíos ni un grupo con todos los elementos.

Por lo tanto, el algoritmo Fuzzy C-Means minimiza la siguiente función objetivo:

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{c} \mu_{ij}^{m} \|x_i - c_j\|^2 \qquad (4)$$

Donde:

- n = número de objetos.
- c = número de centroides.
- m = parámetro fuzzificador, $m > 1$.
- c_j = j-ésimo centroide.
- x_i = i-ésimo objeto.
- $\mu_j(x_i)$ = grado de pertenencia del objeto x_i al j-ésimo grupo.
- $||x_i - c_j||^2$ = La distancia euclidiana entre el objeto x_i y el centroide c_j.
- d = número de dimensiones.

Para hacer el cálculo de las distancias entre objetos y centroides se utilizará la distancia euclidiana, la cual se expresa de la siguiente manera:

$$||x_i - c_j|| = \sqrt{\sum_{d=1}^{D} (c_{jd} - x_{i_d})^2} \qquad (5)$$

Para realizar el cálculo de pertenencia μ_{ij} del objeto x_i al centroide c_j se utiliza la siguiente expresión:

$$\mu_{ij} = \frac{1}{\sum_{c=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_c\|} \right)^{\frac{2}{m-1}}} \quad (6)$$

Donde *m* es el parámetro fuzzificador, $m > 1$. En este caso su valor será *m* = 2. Este valor determina cuán difusa es la clasificación. Usualmente un valor *m* = 2 es seleccionado debido a que permite obtener mejores resultados [11].

Para actualizar los centroides y conocer su nueva posición, se utiliza la siguiente fórmula:

$$c_j = \frac{\sum_{i=1}^{n} \mu_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} \mu_{ij}} \quad (7)$$

Para lograr la convergencia del algoritmo, se establece un criterio de paro. Cuando la diferencia máxima de los objetos, en su posición actual comparada con la posición anterior sea menor a un umbral épsilon (ε) se detiene el algoritmo. Este criterio se puede observar en la Expresión (8).

$$max_{ij} ||\mu_{ij+1} - \mu_{ij}|| < \varepsilon \quad (8)$$

2.2.4. Pseudocódigo y complejidad de Fuzzy C-Means

Las fases del algoritmo Fuzzy C-Means están comprendidas dentro de cuatro etapas las cuales se indican en el extremo derecho de la siguiente lista. Las fases consisten en:

1. Definir los centroides iniciales. } Inicialización
2. Calcular las distancias de cada objeto a cada centroide. }
3. Calcular el grado de pertenencia de cada objeto a cada uno de los centroides. } Clasificación

4. Asignar el objeto al clúster de mayor pertenencia.
5. Actualizar la matriz de pertenencia y recalcular la posición de cada centroide. } Recálculo de centroides
6. Si se alcanza el criterio de parada, terminar, sino volver al paso 2. } Convergencia

La Tabla 1 muestra el pseudocódigo del algoritmo Fuzzy C-Means, en ella se aprecian cada una de las fases mencionadas.

Tabla 1. Pseudocódigo Fuzzy C-Means

Algoritmo Fuzzy C-Means

1	**Inicialización**
2	*N*= $\{x_1, x_2, \ldots, x_n\}$;
3	*C* = $\{c_1, c_2, \ldots, c_c\}$;
4	*m* = parámetro fuzzificador= 2;
5	ε = 0.000050;
6	**Clasificación**
7	Calcular la distancia Euclidiana de cada x_i a los c_j centroides;
8	Calcular μ_{ij} de x_i a c_j;
9	Asignar el objeto x_i al grupo del centroide c_j con mayor grado de pertenencia;
10	**Recalculo de centroides**
11	Actualizar la matriz de pertenencia y recalcular la posición de cada c_j centroide;
12	**Convergencia**
13	Si $max_{ij} \left\|\left\|\mu_{ij+1} - \mu_{ij}\right\|\right\| < \varepsilon$;
14	Detener el algoritmo;
15	En caso contrario:
16	Ir a **Clasificación**
17	Fin del algoritmo

La complejidad de este algoritmo se representa como $O(ndc^2r)$ [18], donde *n* es el número de objetos; *d,* el número de dimensiones; *c,* el total de centroides; y *r,* el número de iteraciones.

2.2.5. Mejoras al algoritmo Fuzzy C-Means

A continuación, se describen algunas investigaciones publicadas y obtenidas de la literatura especializada en las cuales se proponen mejoras al algoritmo Fuzzy C-Means.

- *Reducing the time complexity of the Fuzzy C-Means algorithm* [19]. En esta investigación se desarrolló una eficiente implementación de este algoritmo. Con esta implementación se elimina la matriz de pertenencia, combinando las dos actualizaciones dentro de una sola de los centroides. Con este cambio se afecta significativamente el tiempo de ejecución, debido a que el nuevo algoritmo es lineal con respecto al número de grupos, mientras que el original es cuadrático. Complejidad y calidad: *O(ncd)* donde *n,* es el número de objetos; *c,* es el número de centroides; y *d*, el total de dimensiones. Esta mejora requiere solo 0.5 seg. por iteración, mientras que el método tradicional requiere 2.1seg. Esto significa una mejora de 400%.
- *Improving Fuzzy C-Means clustering based on feature-weight learning* [20]. En este trabajo se propuso un nuevo algoritmo Fuzzy C-Means, basado en una distancia euclidiana ponderada. Esta última incorpora características de peso en la tradicional. Con esta modificación se demuestra que una asignación apropiada de los pesos podrá mejorar el rendimiento de Fuzzy C-Means. El resultado fue un algoritmo denominado WFCM.

  Complejidad y calidad: $O(cn^2)$ donde *n* es el número de objetos y *c* es una constante asociada al número de características.
- *Improving Fuzzy C-Means with Shadow Set* [21]. En esta investigación se propuso un nuevo algoritmo denominado *Shadow Set-Based Fuzzy C-Means (S-FCM* por sus siglas en inglés) con el fin de mejorar la eficiencia y los resultados del agrupamiento. Este algoritmo elimina algunos valores atípicos durante el proceso de agrupamiento y reduciendo el costo de tiempo. S-FCM presenta un mejor rendimiento para enormes conjuntos de datos, especialmente los que no muestran agrupamientos definidos. Para reducir el costo de tiempo, se utilizan conjuntos sombreados que eliminan valores atípicos a través de un umbral óptimo α_i. Al crear un conjunto difuso, cada fila de la matriz de partición representa un clúster.

  Complejidad y calidad: No se reporta la expresión de la calidad, el tiempo de costo promedio utilizando esta mejora es de 0.3997 seg. comparado con 1.374 seg. de Fuzzy C-Means tradicional.
- *An Improved Fuzzy C-Means Clustering Algorithm Based on Shadowed Sets and PSO* [22]. Los autores desarrollaron un algoritmo modificado denominado SP-FCM (por sus siglas en inglés)

basado en la optimización por enjambre de partículas (PSO *Particle Swarm Optimization*) y conjuntos sombreados (*shadowed sets*). Para realizar el agrupamiento, SP-FCM introdujo la propiedad de búsqueda global de PSO a fin de resolver el problema de la convergencia prematura del agrupamiento convencional difuso. Se utilizó la propiedad de equilibrio de vaguedad de conjuntos sombreados para manejar la superposición entre grupos y modelar la incertidumbre en los límites de clase. Este nuevo algoritmo utiliza el índice *Xie-Beni* como validador de grupos y automáticamente encontró el número óptimo de grupos dentro de un rango específico con particiones de grupo que proporcionan grupos compactos y bien separados.

Complejidad y calidad: No se reporta la expresión de la complejidad. Para medir el rendimiento se utilizan los índices DB y Dunn, los cuales en promedio muestran que: con DB 1.005%y con Dunn 2.063, comparado con 1.500 de Fuzzy C-Means tradicional. Esta mejora produce buenos resultados con referencia a los índices DB y Dunn.

- *Improving Fuzzy C-Means clustering algorithm based on a density-induced distance measure* [23]. Se reporta un algoritmo Fuzzy C-Means mejorado en comparación con el tradicional para lo cual se emplea una métrica de distancia inducida por densidad basada en un nuevo método de cálculo de grado de densidad relativa. Mediante el uso de varias instancias sintéticas y reales, el rendimiento del agrupamiento del método propuesto se estudia sistemáticamente y se compara con el método convencional. Los resultados obtenidos apoyan la conclusión de que este nuevo método no solo hereda las buenas características del tradicional, sino que también posee particiones mejoradas. Este nuevo algoritmo se denomina DFCM (*density-induced* Fuzzy C-Means por sus siglas en inglés).

  Complejidad y calidad: No se reporta la expresión de la complejidad. El promedio de la tasa de error es de 11.74% comparado con el 17.34% de Fuzzy C-Means Por lo tanto, se puede concluir que DFCM proporciona un mejor rendimiento.

- *Fuzzy C-Means++: Fuzzy C-Means with effective seeding initialization* [24]. Con esta investigación se comprobó empíricamente la efectividad de la introducción del esquema de inicialización K-Means ++ en el contexto de Fuzzy C-Means con instancias reales. El algoritmo produjo resultados alentadores en términos de reducir la cantidad de iteraciones necesarias para alcanzar la convergencia y la calidad final del *cluster*. Los resultados más importantes se obtuvieron en el medio superpuesto generado y grupos de igual tamaño, donde Fuzzy C-Means ++ fue en promedio 2.1 veces más rápido que el estándar.

Complejidad y calidad: No se reporta la expresión de la complejidad. En promedio, Fuzzy C-Means ++, se desempeña mejor que el enfoque tradicional en grupos desiguales y superpuestos, necesitando en promedio 86.8 iteraciones, mientras que el algoritmo tradicional necesita 111.

- *Improving fuzzy c-mean-based community detection in social networks using dynamic parallelism* [25]. En este artículo, se presentaron diferentes y novedosas implementaciones de la tarjeta GPU en ambas versiones del algoritmo Hybrid CPU-GPU (Dynamic Parallel & Hybrid Nested Parallel). Con los resultados obtenidos se mejoró el rendimiento tanto de Fuzzy C-Means como de K-Means mediante el uso de unidades de procesamiento gráfico GPU.

  Complejidad y calidad: No se reporta la expresión de la complejidad. El promedio del tiempo de ejecución con la implementación paralela fue: DP 6.9602, HCG 3.3277, HNP 2.215. Comparado con el secuencial Fuzzy C-Means 26.245%, se demostró que es mejor el rendimiento usando paralelismo.

Con la mejora que se propone al algoritmo Fuzzy C-Means a partir de esta investigación, se podrá comparar estas mejoras. Se puede observar que son equiparables, ya que hasta ahora no se ha aplicado alguna mejora a la fase de convergencia del algoritmo, con lo cual se ahorra un gran porcentaje de tiempo y se disminuye el número de iteraciones necesarias para converger sin pérdida significativa de la calidad.

2.2.6. Heurística O-K-Means

En la investigación de Almanza [9], se propuso un criterio de paro que permite hacer un balance entre el tiempo de procesamiento y la calidad de la solución del algoritmo K-Means cuando se solucionan instancias con un número de n objetos muy grande. Hasta ese momento, la mayoría de las mejoras propuestas al algoritmo se enfocaban en las fases de inicialización y clasificación. En contraste, O-K-Means se aplicó a la fase de convergencia, en la cual el criterio de paro consistió en detener el algoritmo cuando el número de objetos que cambia de grupo es menor a un umbral definido.

Derivado de una intensiva experimentación computacional, que se realizó para analizar el comportamiento y las tendencias del algoritmo K-Means bajo diferentes condiciones, al aplicar O-K-Means, se observó una correlación entre los valores de la función objetivo y el número de objetos que cambian de grupo por iteración. Esta heurística detiene el algoritmo cuando el total de objetos que cambian de grupo (γ) en una iteración es menor a un umbral dado (U). Para determinar el valor más adecuado para el umbral, la autora aplicó el principio de Pareto, que define una relación óptima entre esfuerzo computacional y beneficio en la calidad de la solución.

A continuación, se enlistan los principales criterios de interés de dicha heurística, los cuales motivaron que fuera seleccionada para mejorar el algoritmo Fuzzy C-Means:

- Se aplicó a la fase de convergencia.
- Se propuso un valor del umbral de 0.72% del total de los objetos.
- No requirió memoria adicional.

En la siguiente lista se observan los resultados obtenidos en esta investigación:

- Reducción de tiempo promedio un 93.88% en instancias sintéticas y 88.21% para instancias reales.
- Reducción promedio de la calidad de 0.40% en instancias sintéticas y de 0.20% en instancias reales.
- Redujo significativamente el número de iteraciones requeridas por el algoritmo K-Means para converger con una disminución de la calidad relativamente pequeña.
- Recomendada cuando el usuario prioriza la calidad de la solución.

En la Tabla 2, se muestra el pseudocódigo de O-K-Means, en el cual se incorpora el criterio de paro.

Tabla 2. Pseudocódigo algoritmo O-K-Means

Algoritmo O-K-Means

1	**Inicialización**
2	$N = \{x_1, \ldots, x_n\}$;
3	$M = \{\mu_1, \ldots \mu_k\}$
4	U = Valor del umbral;
5	**Clasificación**
6	Para $x_i \in N$ y $\mu k \in M$ {
7	Calcular la distancia Euclidiana de cada x_i a los k centroides;
8	Asignar el objeto x_i al centroide μ_k más cercano;
9	Calcular γ;}
10	**Cálculo de centroides**
11	Calcular el centroide μ_k;
12	**Convergencia**
13	Sí ($\gamma \leq U$);
14	Detener el algoritmo;
15	En caso contrario;
16	Ir a **Clasificación**
17	Fin del algoritmo

En la Tabla anterior, se describe el pseudocódigo del algoritmo O-K-Means. En la etapa de Inicialización, se requiere conocer el conjunto de objetos y de centroides, así como el valor de un umbral que será determinado aplicando el Principio de Pareto. Para la fase de Clasificación se realizará el cálculo de la distancia euclidiana entre cada objeto y cada centroide. De esta manera, cada objeto se asignará al centroide más cercano, con lo cual se podrá hacer el cálculo del porcentaje de objetos que cambian de grupo (γ). Se aplica $100(u_i/n)$ donde u_i es el número de objetos cambiando de grupo. Se hace el cálculo de centroides, y si se cumple el criterio de paro (Convergencia) el cual consiste en que, si el porcentaje de objetos cambiando de grupo es menor o igual a un umbral dado, el algoritmo se detendrá, si nó, se regresará a la fase de Clasificación.

Después de describir y analizar el proceso anterior, se llegó a la conclusión de integrar el algoritmo Fuzzy C-Means con la heurística O-K-Means, debido a que, el algoritmo es de agrupamiento y tiene características similares a K-Means; y la heurística mostró mejorar de manera significativa el tiempo de ejecución del algoritmo antes mencionado sin una pérdida significativa de su calidad. Lo que abrió la posibilidad de experimentar con un algoritmo diferente y comprobar si funciona de igual manera.

Capítulo 3

Integración y adecuación del algoritmo Fuzzy C-Means con la heurística O-K-Means

El humor y la curiosidad son la más pura forma de inteligencia.
Roberto Bolaño

Una vez que se estudiaron los algoritmos de agrupamiento y las heurísticas orientadas a la mejora de K-Means, se realizó la caracterización de cada uno de ellos. Como resultado se seleccionó el algoritmo Fuzzy C-Means, el que se pretende mejorar. Asimismo, se seleccionó la heurística O-K-Means, que se consideró adecuada para dicho propósito.

Para llevar a cabo el proceso de integración y adecuación de una heurística es importante conocer con precisión el comportamiento del algoritmo que se pretende mejorar. Con base en los hallazgos realizados en la revisión de las aportaciones contenidas en el estado del arte, se tomó la decisión de mejorar el algoritmo de agrupamiento Fuzzy C-Means mediante la heurística desarrollada por Almanza [9] denominada *Optimizaded*-K-Means (O-K-Means), con base en los siguientes criterios:

Para el caso del algoritmo:

- Que pertenezca a la familia de los algoritmos de agrupamiento.
- Que se derive del algoritmo K-Means.
- Que fuera de uso generalizado dentro de la comunidad científica.
- Que sea de fácil aplicación, pero que presente un costo computacional alto.

Para el caso de la heurística:

- Que fuera desarrollada en el CENIDET.
- Que, a diferencia de otras heurísticas, las que en su mayoría se aplican a las fases de inicialización y clasificación de K-Means, ésta se aplique en la fase de convergencia.
- Que utilice el Principio de Pareto para determinar un umbral y proponer un nuevo criterio de paro, reduciendo el costo del algoritmo. Esta es una característica novedosa de la heurística seleccionada.

3.1. Proceso de experimentación con Fuzzy C-Means

Para iniciar este proceso de experimentación, se llevó a cabo la programación computacional del algoritmo y la heurística seleccionados. Para realizarlo se utilizaron los siguientes recursos computacionales:

- La codificación del algoritmo y su mejora se desarrollaron con el lenguaje de programación C.
- Se utilizó el sistema operativo *Linux*, distribución *Ubuntu* 18.04.
- El equipo de cómputo fue una laptop *Dell Inspiron* 15 7000, 1 Tb de disco duro, 8 Gb de RAM, procesador *Intel Core* i7 de 7° generación a 2.80 GHz.

Las instancias reales utilizadas en esta fase, se tomaron del repositorio *UCI Machine Learning* [26], las cuales se describen en la Tabla 3:

Tabla 3. Instancias utilizadas para la fase de experimentación

Nombre de la instancia	***n***	***d***	***k***
Iris	150	4	3,4,5
Breast cáncer	683	9	3,4,5
Wine	178	13	3,4,5
Ecoli	336	7	3,4,5
RSSI	1420	13	3,4,5
Power Plant	9568	5	3,4,5

En la Tabla 3, se observan las diferentes instancias utilizadas para esta investigación, las cuales fueron obtenidas del repositorio de la Universidad *Irvine* de California, todas contienen sólo datos numéricos y están separados por comas. La primera columna indica el nombre de la instancia; en la segunda columna, el total de objetos; en la tercera, se encuentra el número de dimensiones de la instancia *n*; y en la cuarta columna se incluye el número de grupos con los que se experimentó en cada instancia.

Con cada instancia, se realizó una experimentación computacional con 3, 4 y 5 grupos respectivamente. En la siguiente sección, se presentará una descripción detallada del proceso y los resultados de la ejecución de la instancia *Ecoli*.

3.2. Resultados de la ejecución del algoritmo Fuzzy C-Means

A continuación, se presentan los resultados obtenidos a partir de la ejecución del algoritmo Fuzzy C-Means con la instancia *Ecoli*. Esta instancia se eligió como ejemplo para describir los resultados, debido a que sus datos se pueden representar con mayor claridad.

La Tabla 4 está compuesta por cuatro columnas: el número de iteración *r*; el valor de la función objetivo z_r; los objetos que cambiaron de grupo v_r; y el criterio de paro de Fuzzy C-Means. Como se podrá apreciar, para la instancia *Ecoli* se realizaron un total de 212 iteraciones. Con base en esta observación, sobresalen los siguientes aspectos:

Como se ha visto en investigaciones anteriores, durante la ejecución de *Ecoli*, los objetos cambian constantemente de grupo. Con esta experimentación, a partir de la iteración 21 los objetos ya no cambiaron de grupo. Sin embargo, el algoritmo continuó iterando porque no se había cumplido el criterio de paro establecido, llegando a un total de 212 iteraciones.

Tomando en cuenta los resultados obtenidos, se concluyó que es posible mejorar el algoritmo Fuzzy C-Means, cuando se optimiza el criterio de paro. Es importante recordar, para que el algoritmo se detenga, el criterio de paro consiste en aplicar la condición de que el valor de la diferencia de la función de membresía debe ser menor al valor de épsilon (ε), el cual estuvo establecido desde la etapa de inicialización.

Tabla 4. Resultados de la ejecución de la instancia *Ecoli* con el algoritmo Fuzzy C-Means

Iteración r	**Función objetivo** z_r	**Objetos que cambiaron de grupo** v_r	**Criterio de paro** ε**=0.000050**
1	29.06084	0	0.32786
2	26.365739	6	0.404702
3	14.592427	15	0.494925
4	10.547772	16	0.319022
5	9.599315	13	0.15347
6	9.332565	6	0.212465
7	9.263006	4	0.074718
8	9.242968	0	0.039072
9	9.234756	4	0.0244
10	9.230566	0	0.02189
11	9.228232	3	0.018257
12	9.226899	1	0.014452
13	9.22614	0	0.011012
14	9.225708	0	0.00817
15	9.225459	3	0.005955
16	9.22531	0	0.004299
17	9.225215	0	0.003092
18	9.225151	0	0.002323
19	9.225105	0	0.002068
20	9.225068	0	0.001888
21	9.225039	1	0.001727
22	9.225015	0	0.001585
23	9.224994	0	0.00146
24	9.224975	0	0.001349
25	9.224959	0	0.00131
…	…	…	…
175	9.224767	0	0.000075
176	9.224767	0	0.000075
177	9.224767	0	0.000074
178	9.224767	0	0.000073
179	9.224767	0	0.000072
180	9.224767	0	0.000071
181	9.224767	0	0.00007
182	9.224767	0	0.000069
183	9.224767	0	0.000069
184	9.224767	0	0.000068
185	9.224767	0	0.000067
…	…	…	…
202	9.224767	0	0.000055
203	9.224767	0	0.000055
204	9.224767	0	0.000054
205	9.224767	0	0.000054
206	9.224767	0	0.000053
207	9.224767	0	0.000053
208	9.224767	0	0.000052
209	9.224767	0	0.000051
210	9.224767	0	0.000051
211	9.224767	0	0.00005
212	9.224767	0	0.00005

3.3. Correlación entre objetos que cambian de grupo y función objetivo

En esta sección, se analizará la correlación que se presenta entre los objetos que cambian de grupo y el valor de la función objetivo durante la ejecución desarrollada con el algoritmo Fuzzy C-Means. En la Tabla 5, se muestra solamente una parte de los resultados obtenidos al solucionar la instancia *Ecoli* aplicando la metodología desarrollada por Almanza [9]. El algoritmo se detuvo en la iteración 212 cuando se cumplió el criterio de paro y además, los objetos ya no cambiaron de grupo. En dicha tabla, cada fila contiene información de cada iteración *r*, y del valor de la función objetivo (z_r).

Se denota que z^* es el valor más bajo encontrado por el algoritmo, esto es $z^* = z_{212} = 9.224767$. El porcentaje de objetos que cambian de grupo está representado como $\gamma_r = 100(v_r/n)$. Donde v_r es el número de objetos cambiando de grupo. Por otra parte, el porcentaje de la función objetivo $\delta_r = 100(z_r/z^* - 1)$ expresa la diferencia entre z_r y z^* como un porcentaje de z^*. [9].

Tabla 5. Resultados de la instancia *Ecoli* con porcentaje de objetos cambiando de grupo y diferencias de la función objetivo

r	z_r	γ_r	δ_r
1	29.06084	0	215.0306127
2	26.365739	1.785714286	185.8146878
3	14.592427	4.464285714	58.18748593
4	10.547772	4.761904762	14.34187985
5	9.599315	3.869047619	4.060243473
6	**9.332565**	**1.785714286**	**1.168571521**
7	9.263006	1.19047619	0.414525375
8	9.242968	0	0.197305796
9	9.234756	1.19047619	0.108284578
10	9.230566	0	0.062863376
11	9.228232	0.892857143	0.037561924
12	9.226899	0.297619048	0.023111695
13	9.22614	0	0.014883845
14	9.225708	0	0.0102008
15	9.225459	0.892857143	0.007501544
16	9.22531	0	0.005886328
17	9.225215	0	0.004856491
18	9.225151	0	0.004162707
19	9.225105	0	0.003664049
20	9.225068	0	0.003262955
21	9.225039	0.297619048	0.002948584
22	9.225015	0	0.002688415
23	9.224994	0	0.002460767
24	9.224975	0	0.002254799
25	9.224959	0	0.002081353
…	…	…	…
175	9.224767	0	0
176	9.224767	0	0
177	9.224767	0	0
178	9.224767	0	0
179	9.224767	0	0
180	9.224767	0	0
181	9.224767	0	0
182	9.224767	0	0

183	9.224767	0	0
184	9.224767	0	0
185	9.224767	0	0
…	…	…	…
202	9.224767	0	0
203	9.224767	0	0
204	9.224767	0	0
205	9.224767	0	0
206	9.224767	0	0
207	9.224767	0	0
208	9.224767	0	0
209	9.224767	0	0
210	9.224767	0	0
211	9.224767	0	0
212	9.224767	0	0

Para comprender el comportamiento del algoritmo a partir de los datos de la Tabla 5, en la Figura 4 se muestran de manera gráfica los puntos (γ_r, δ_r) correspondientes a las iteraciones r = 5, …, 25. Nótese una relación cuasi-lineal.

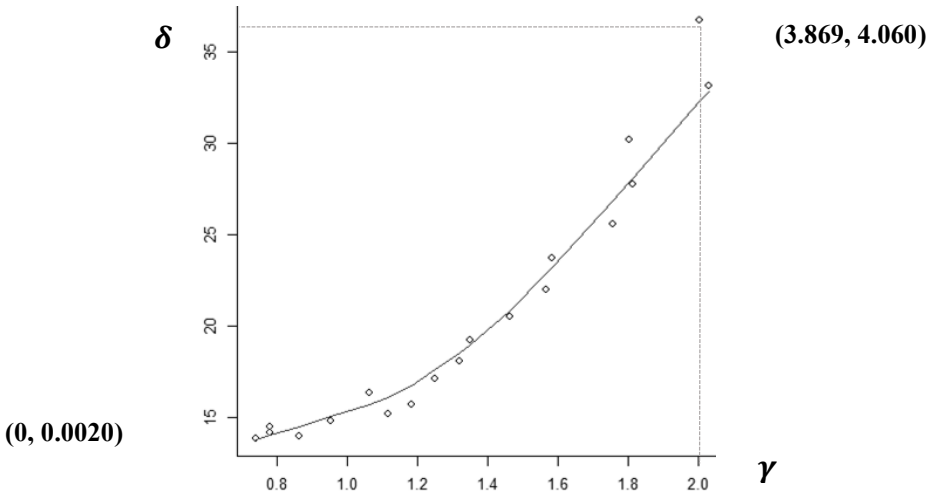

Figura 4. Relación cuasi-lineal para mostrar la correlación entre γ y δ

Sólo se tomaron los datos de las iteraciones 5 a la 25, como una muestra del total de las 212 iteraciones con el propósito de ejemplificar la relación cuasi-lineal entre γ_r (porcentajes de objetos que cambian de grupo) y δ_r (porcentaje de la función objetivo). Esta muestra puede observarse en la Tabla 7.

Para comprobar esta observación, se utilizó el coeficiente de correlación de Pearson, el cual es una medida lineal que puede utilizarse para medir el grado de relación de dos variables, se representa con la Ecuación 9.

$$\rho(\gamma,\delta)=\frac{\sum_{i=2}^{\ell}[(\gamma_i-\bar{\gamma})(\delta_i-\bar{\delta})]}{\sqrt{\sum_{i=2}^{\ell}(\gamma_i-\bar{\gamma})^2}\ \sqrt{\sum_{i=2}^{\ell}(\delta_i-\bar{\delta})^2}} \tag{9}$$

Donde γ es el porcentaje de objetos cambiando de grupo; δ el porcentaje de la disminución de la función objetivo; i el número de iteración y ℓ es el número total de iteraciones. El resultado de dicha correlación puede tomar valores entre -1 y +1 pasando por cero. Esto quiere decir que, entre más cercano esté el valor a +1, la relación es casi perfecta, por el contrario, si está más cercana a cero, la relación la relación es más débil o nula.

Para el ejemplo anterior, se calculó el coeficiente de correlación entre los índices γ y δ y se obtuvo un valor de ρ =0.77, lo cual comprueba la existencia de una correlación fuerte.

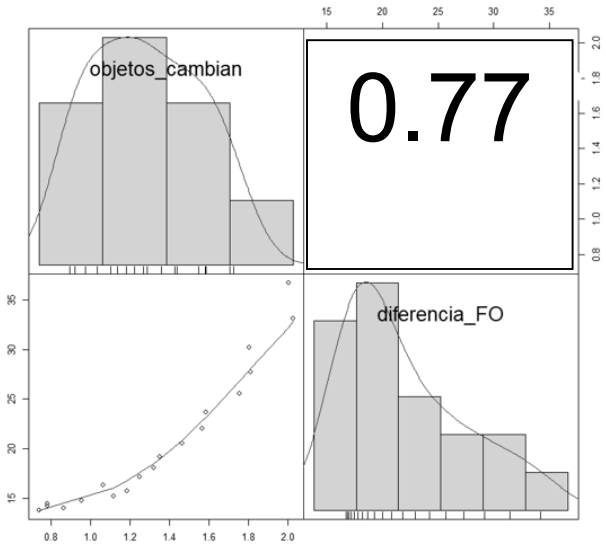

Figura 5. Correlación entre γ y δ

Una vez que se identificó la existencia de una fuerte correlación entre índices, fue posible utilizar la estrategia de sustitución de variables. En este caso, se manejaron como índices. Con esta base, se propone un nuevo criterio de paro basado en un umbral *U*. En la siguiente sección, se describe el proceso para definir dicho umbral de manera óptima.

3.4. Propuesta de mejora con la heurística OFCM

Con base en las experimentaciones realizadas con la instancia *Ecoli* para la ejecución del algoritmo Fuzzy C-Means, se observó que existe una alta correlación entre el valor de la función objetivo y el número de objetos cambiando de grupo. Por esta razón, fue posible proponer una heurística para la mejora a dicho algoritmo denominada *Optimizaded* Fuzzy C-Means OFCM, la cual consiste en un nuevo criterio de paro que se establece con un umbral aplicado a los objetos que cambian de grupo en una iteración dada.

La complejidad de OFCM está representada por: $O(nc^2dr\alpha)$. Dónde: n es el número de objetos, c el número de centroides, d es el número de dimensiones, r el número de iteraciones de Fuzzy C-Means y α el cociente de dividir las iteraciones de OFCM entre las de Fuzzy C-Means.

Para determinar el valor de este umbral se decidió aplicar el Principio de Pareto, el cual define una relación óptima entre el esfuerzo computacional y el beneficio de la calidad en cuanto a la solución de diferentes instancias. Se llegó a la conclusión de que, si se aplica esta heurística al algoritmo Fuzzy C-Means se puede reducir su complejidad.

Con las experimentaciones realizadas se observa que, para que el algoritmo logre converger se necesita cumplir con el criterio de paro, el cual establece que, si el número de objetos cambiando de grupo es menor o igual al valor de épsilon, el algoritmo se detendrá. Esto genera dos situaciones: la primera es que, aunque en determinado número de iteraciones ya no hay objetos cambiando de grupo, estas iteraciones continúan debido a que aún no se ha cumplido el criterio de paro, consumiendo así más recursos. La segunda es contraparte de esta situación, el algoritmo converge aun cuando haya objetos cambiando de grupo, produciendo un resultado no óptimo.

El algoritmo OFCM propone un nuevo criterio de paro, el cual es $\gamma \leq U$. Es decir, cuando el porcentaje de objetos cambiando de grupo es menor o igual a un umbral dado el algoritmo converge. En la siguiente sección se describe como obtener este valor; también se observa que existe una correlación entre el porcentaje de objetos cambiando de grupo y el porcentaje acumulado de la función objetivo.

La implementación de esta heurística fue relativamente sencilla, ya que no requiere de memoria adicional, únicamente lleva el registro de los objetos que cambian de grupo, diferenciándola y otorgándole ventaja sobre otros criterios propuestos. Además, proporciona un parámetro que permite regular la relación entre esfuerzo y calidad de la solución.

Tabla 6. Descripción del pseudocódigo *Optimizaded Fuzzy C-Means*

Algoritmo OFCM

1	**Inicialización**
2	$N = \{x_1, \ldots, x_n\}$;
3	$C = \{c_1, c_2 \ \ldots \ c_c\}$
4	U = Valor del umbral;
5	**Clasificación**
6	Calcular la distancia Euclidiana de cada x_i a los c_j centroides;
7	Calcular μ_{ij} de x_i a c_j;
8	Asignar el objeto x_i al centroide c_j con mayor grado de pertenencia;
9	Calcular γ;}
10	**Cálculo de centroides**
11	Actualizar la matriz de pertenencia y recalcular la posición de cada c_j centroide;
12	**Convergencia**
13	Sí ($\gamma \leq U$);
14	Detener el algoritmo;
15	En caso contrario;
16	Ir a **Clasificación**
17	Fin del algoritmo

En la Tabla anterior, se observa que para iniciar la ejecución del algoritmo OFCM se necesita del conjunto de objetos, centroides y el valor de un umbral, el cual se obtiene aplicando el principio de Pareto. Para entrar a la fase de clasificación se debe calcular la distancia euclidiana de cada objeto a cada centroide, asignar el objeto al centroide con mayor grado de pertenencia y calcular el porcentaje de objetos cambiando de grupo de acuerdo con la metodología desarrollada por Almanza [9]. Se hará el recalculo de los centroides y si se determina que se cumple el criterio de paro, cuando el porcentaje de objetos cambiando de grupos es menor o igual a un umbral dado, el algoritmo se detendrá, si no, regresará a la fase de clasificación.

3.5. Determinando el valor del umbral

Como fue posible observar durante la ejecución de Fuzzy C-Means, es de destacar la fuerte correlación que existe entre el valor de la función objetivo y el número de objetos cambiando de grupo. Este resultado lleva a la siguiente pregunta ¿Cuántas iteraciones son necesarias para converger? Para resolver esta

cuestión, se propone determinar el valor de un umbral con el fin de establecer un criterio de paro y también, plantear una heurística basada en la observación antes mencionada.

Para poder establecer el criterio de paro, primeramente, se debe determinar el valor del umbral que servirá para indicar al algoritmo cuando es conveniente converger.

Continuando con el ejemplo de los resultados de la ejecución de Fuzzy C-Means, se calculó, el porcentaje de iteraciones A_r; el porcentaje de reducción de la función objetivo B_r; el porcentaje acumulado de la calidad C_r; y la distancia euclidiana entre los puntos (A_r, C_r) y (0,100) D_r. Lo cual se representa con la siguiente ecuación:

$$
\begin{aligned}
&r \geq 2, \\
&A_r = 100(r/\ell) \\
&B_r = 100(z_{r-1} - z_r)/(z_1 - z^*) \\
&C_r = C_{r-1} + B_r \\
&D_r = \sqrt{(0 - A_r)^2 + (100 - C_r)^2}
\end{aligned}
\qquad (10)
$$

Donde:

$r =$ Iteración

$\ell =$ Número total de iteraciones

$z =$ Valor de la función objetivo

$z^* =$ Valor más bajo de la función objetivo

$A_r =$ Porcentaje acumulado de iteraciones

$B_r =$ Porcentaje acumulado de la reducción de z

$C_r =$ Porcentaje de mejora de calidad

$D_r =$ Distancia euclidiana entre los puntos (A_r, C_r) y (0, 100)

En la Tabla 7, se muestran los resultados obtenidos con la ejecución de la instancia *Ecoli* con Fuzzy C-Means, los cuales representan la relación entre el esfuerzo computacional y la calidad de solución con la instancia *Ecoli*. Estos resultados están expresados en porcentajes. En la primera columna se observa el número de iteración; en la segunda el porcentaje de iteraciones; en la tercera el porcentaje acumulado de

la reducción del valor de la función objetivo; en la cuarta columna se muestra el porcentaje de la mejora de la calidad; y en la última columna, la distancia euclidiana entre los puntos (A_r, C_r) y (0, 100)

Tabla 7. Relación entre el esfuerzo computacional y la calidad de solución con la instancia *Ecoli*

r	A_r	B_r	C_r	D_r
1	0.471698113			
2	0.943396226	13.58686772	13.58686772	86.41828178
3	1.41509434	44.65382897	58.24069669	41.78327302
4	1.886792453	27.71749346	85.95819016	14.16800654
5	2.358490566	8.992012721	94.95020288	5.573412663
6	2.830188679	2.778844115	97.72904699	3.628663047
7	3.301886792	0.745336357	98.47438335	3.637301547
8	3.773584906	0.216322865	98.69070622	3.994270052
9	4.245283019	0.08884592	98.77955214	4.417230002
10	4.716981132	0.04537207	98.82492421	4.861143293
11	5.188679245	0.025285557	98.85020976	5.314547008
12	5.660377358	0.014444804	98.86465457	5.773117103
13	6.132075472	0.008225949	98.87288052	6.234801354
14	6.603773585	0.004682348	98.87756287	6.698484208
15	7.075471698	0.00269898	98.88026185	7.163526596
16	7.547169811	0.001615096	98.88187694	7.629545946
17	8.018867925	0.001029776	98.88290672	8.096304107
18	8.490566038	0.000693751	98.88360047	8.563647561
19	8.962264151	0.000498637	98.8840991	9.03146796
20	9.433962264	0.000401079	98.88450018	9.499683355
21	9.905660377	0.000314361	98.88481455	9.968236861
22	10.37735849	0.000260162	98.88507471	10.43707946
23	10.8490566	0.000227642	98.88530235	10.90617165
24	11.32075472	0.000205962	98.88550831	11.37548149
25	11.79245283	0.000173442	98.88568175	11.84498412
…	…	…	…	…
175	82.54716981	0	98.88776308	82.55466259
176	83.01886792	0	98.88776308	83.02631813
177	83.49056604	0	98.88776308	83.49797416
178	83.96226415	0	98.88776308	83.96963065
179	84.43396226	0	98.88776308	84.44128762
180	84.90566038	0	98.88776308	84.91294504
181	85.37735849	0	98.88776308	85.38460291
182	85.8490566	0	98.88776308	85.85626122
183	86.32075472	0	98.88776308	86.32791997
184	86.79245283	0	98.88776308	86.79957914
185	87.26415094	0	98.88776308	87.27123874
…	…	…	…	…
202	95.28301887	0	98.88776308	95.28951021
203	95.75471698	0	98.88776308	95.76117635
204	96.22641509	0	98.88776308	96.2328428
205	96.69811321	0	98.88776308	96.70450956
206	97.16981132	0	98.88776308	97.17617662
207	97.64150943	0	98.88776308	97.64784399
208	98.11320755	0	98.88776308	98.11951165
209	98.58490566	0	98.88776308	98.5911796
210	99.05660377	0	98.88776308	99.06284784
211	99.52830189	0	98.88776308	99.53451636
212	100	0	100	100

La Figura 6, muestra una gráfica parcial con los resultados de la aplicación del diagrama de Pareto para las iteraciones 5, 6 y 7 y muestra los puntos (A_r, C_r) tomados de la Tabla 7. El objetivo fue identificar el punto de inflexión dónde se encuentre el menor esfuerzo sin mayor pérdida de calidad. Ese punto es el que tiene menor distancia al punto (0, 100).

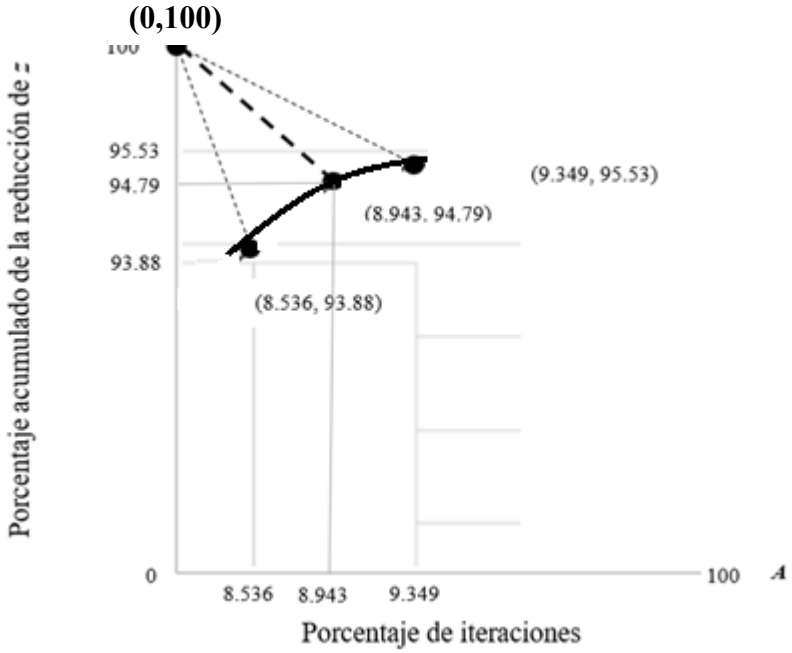

Figura 6. Muestra parcial del porcentaje de iteraciones y de la calidad aplicando el diagrama de Pareto

En la Figura anterior, el eje x muestra el porcentaje total de iteraciones que realizó el algoritmo. En este caso se muestra un intervalo de 3.8690 a 1.190 que corresponde a las iteraciones r =5, 6 y 7. El eje y muestra el porcentaje acumulado de la reducción de la función objetivo en cada iteración. De la misma manera, se muestra un intervalo de 4.060 a 0.414 que corresponde a las iteraciones r =5, 6 y 7. Nótese que $D_6 \leq D_r$ para toda $r \in \ell$, es decir, (A_6, C_6) , es decir en la iteración 6, el punto más cercano a (0, 100). Por lo tanto, se puede detener el algoritmo Fuzzy C-Means en la iteración r = 6 = 2.830 ℓ, evitando el 94% de las iteraciones, para obtener una solución con un costo de z_6=9.332 ($\delta_6 = 1.168$ como se muestra en la Tabla 8). Además, como se observa en esta misma Tabla, en la iteración 6 migran tan sólo el 1.78% objetos.

Tabla 8. Cambio de objetos por iteración de la instancia *Ecoli*

r	Número de objetos que cambian de grupo	Porcentaje de objetos que cambian de grupo
1	0	0
2	6	1.78
3	15	4.46
4	16	4.76
5	13	3.86
6	**6**	**1.78**
7	4	1.19
8	0	0
9	4	1.19
10	0	0

Por lo anterior, siguiendo los objetivos y conceptos presentados, es razonable detener el algoritmo Fuzzy C-Means cuando $\gamma_r \leq U$, es decir, cuando el porcentaje de objetos cambiando de grupo es menor o igual al umbral de objetos definido. En particular, en el ejemplo que se desarrolló, se obtuvo $U = 178$ (ver Tabla 8, iteración 6, resaltada con negritas).

Capítulo 4

Pruebas para la mejora de OFCM y análisis de resultados

¡Eureka! (¡Lo he descubierto!)
Arquímedes

Una vez realizadas las experimentaciones con las ejecuciones de Fuzzy C-Means y de OFCM, en este capítulo se describirán las pruebas realizadas y los análisis de los resultados obtenidos.

4.1. Descripción de los casos de prueba

De acuerdo con la literatura especializada [9], se ha demostrado que el uso de un tamaño de muestra mayor o igual a 30 cumple con el rigor para realizar una inferencia estadística. Las instancias utilizadas son las mismas mencionadas en la Tabla 3 (ver página 26).

En la Tablas 9, 10 y 11, se presentan los resultados de cada instancia que muestran el valor de la función objetivo y el número de iteraciones con Fuzzy C-Means; así como el valor del umbral propuesto y el número de iteraciones necesarias para llegar a dicho umbral, para $k = 3$, 4 y 5 grupos respectivamente. La primera columna indica el nombre de la instancia, la segunda y tercera columnas muestran el valor más bajo de la función objetivo (z^*) y el total de iteraciones con el algoritmo Fuzzy C-Means. Las dos últimas columnas muestran el valor del umbral y el número de iteraciones para ese umbral.

Mediante el análisis de los resultados que se presentan en las tablas anteriores, se observan porcentajes de ahorro de iteraciones en todas las instancias. También se observó que los resultados más notorios se dan en aquellas instancias que contienen mayor cantidad de objetos, es decir, las más grandes y cuando $k = 5$; es importante hacer esta aclaración debido a que la heurística seleccionada para mejorar el algoritmo está orientada para trabajar con grandes volúmenes de datos o *Big Data*.

En los experimentos descritos, para cada instancia se utilizó un valor diferente del umbral U de acuerdo con el resultado obtenido de la aplicación del principio de Pareto.

Tabla 9. Resultados que muestran el valor de la función objetivo, el número de iteraciones y el valor de umbral k=3

Instancia	**Fuzzy C-Means**		***Optimized* Fuzzy C-Means**	
	z^*	**Iteraciones**	**Valor del umbral**	**Iteraciones**
Iris	69.949523	18	1.33	6
Wine	94.743188	18	1.12	4
Ecoli	9.789658	26	4.76	4
Breast	5417.157422	22	1.02	4
Power Plant	548506.9697	39	5.37	6
RSSI	300013.1451	40	0.42	3

Tabla 10. Resultados que muestran el valor de la función objetivo, el número de iteraciones y el valor de umbral k=4

Instancia	**Fuzzy C-Means**		***Optimizaded* Fuzzy C-Means**	
	z^*	**Iteraciones**	**Valor del umbral**	**Iteraciones**
Iris	67.101999	43	2	6
Wine	90.43581	29	1.12	5
Ecoli	9.593981	31	4.16	3
Breast	5409.042297	19	1.75	3
Power Plant	542368.9496	42	4.17	7
RSSI	273014.3917	57	0.07	5

Tabla 11. Resultados que muestran el valor de la función objetivo, el número de iteraciones y el valor de umbral k=5

Instancia	**Fuzzy C-Means**		***Optimizaded* Fuzzy C-Means**	
	z^*	**Iteraciones**	**Valor del umbral**	**Iteraciones**
Iris	66.528855	82	1.33	6
Wine	87.99513	88	1.12	5
Ecoli	9.224767	212	1.78	6
Breast	5404.603689	21	2.19	4
Power Plant	539233.7329	39	7.90	5
RSSI	248459.8698	52	0.07	18

Nótese que existe una disminución considerable de la cantidad de iteraciones entre el algoritmo Fuzzy C-Means y OFCM. En las iteraciones es más notable la reducción de esfuerzo, ya que Fuzzy C-Means realiza un número mayor de iteraciones que OFCM. Por lo tanto, es posible afirmar que es factible aplicar el criterio de paro propuesto para mejorar un algoritmo de agrupamiento, en este caso, Fuzzy C-Means.

A continuación, en las Figuras 7, 8 y 9 se muestran en forma comparativa la diferencia del total de iteraciones entre los algoritmos Fuzzy C-Means y OFCM con una corrida por cada una de las instancias para k= 3, 4 y 5.

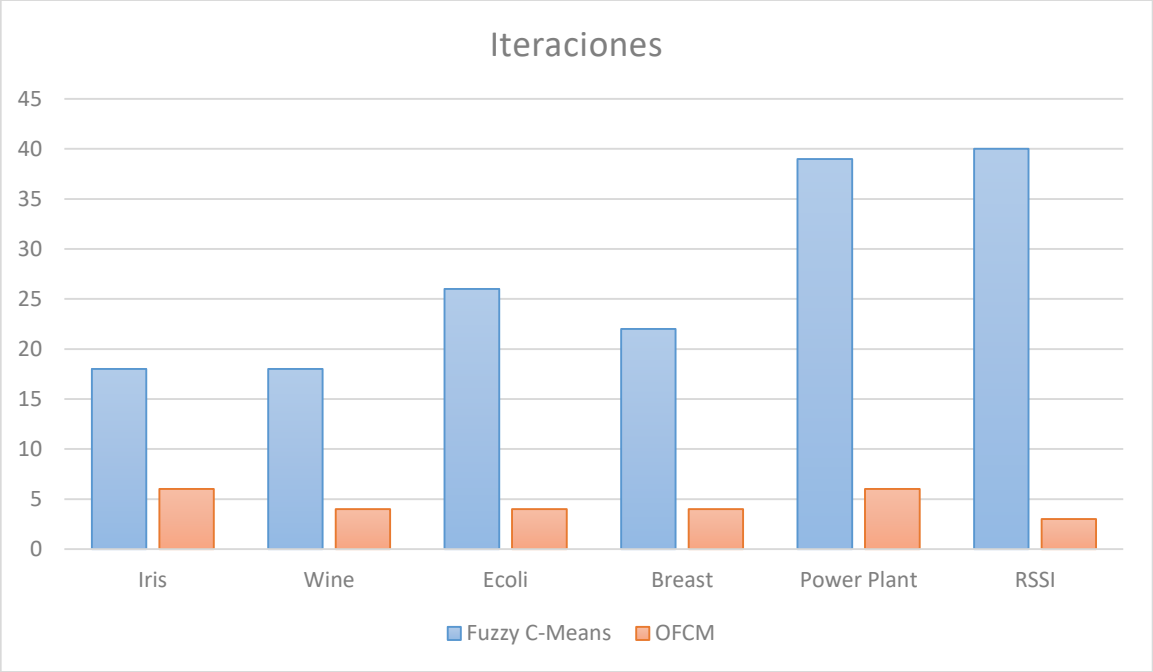

Figura 7. Total de iteraciones entre Fuzzy C-Means y OFCM k=3

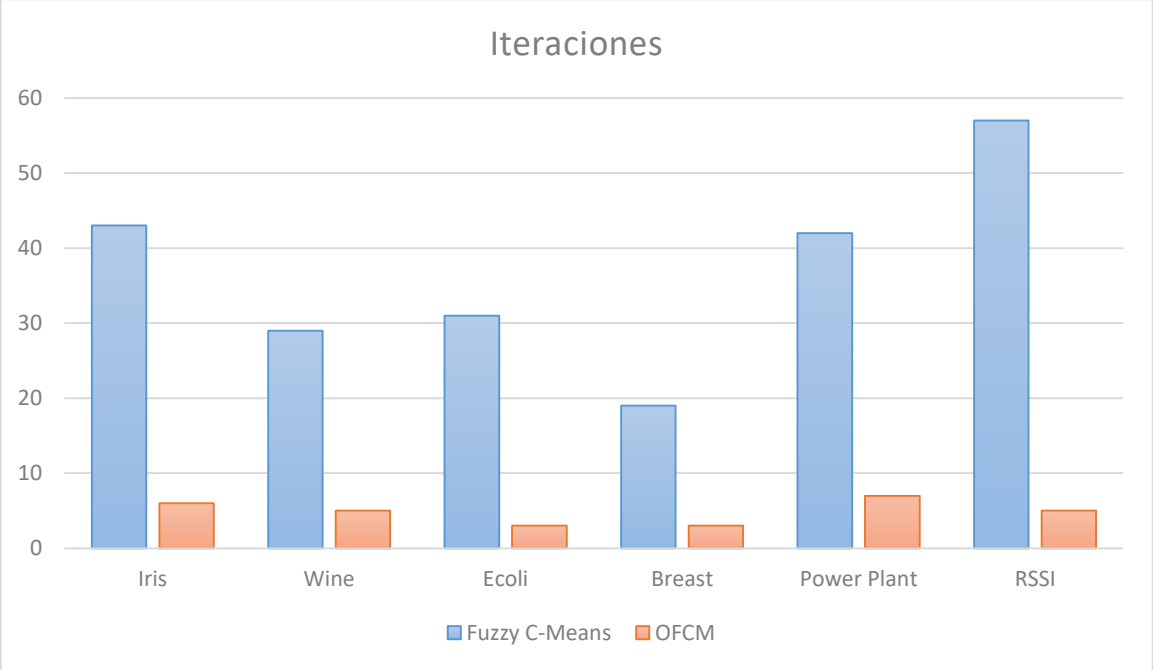

Figura 8. Total de iteraciones entre Fuzzy C-Means y OFCM k=4

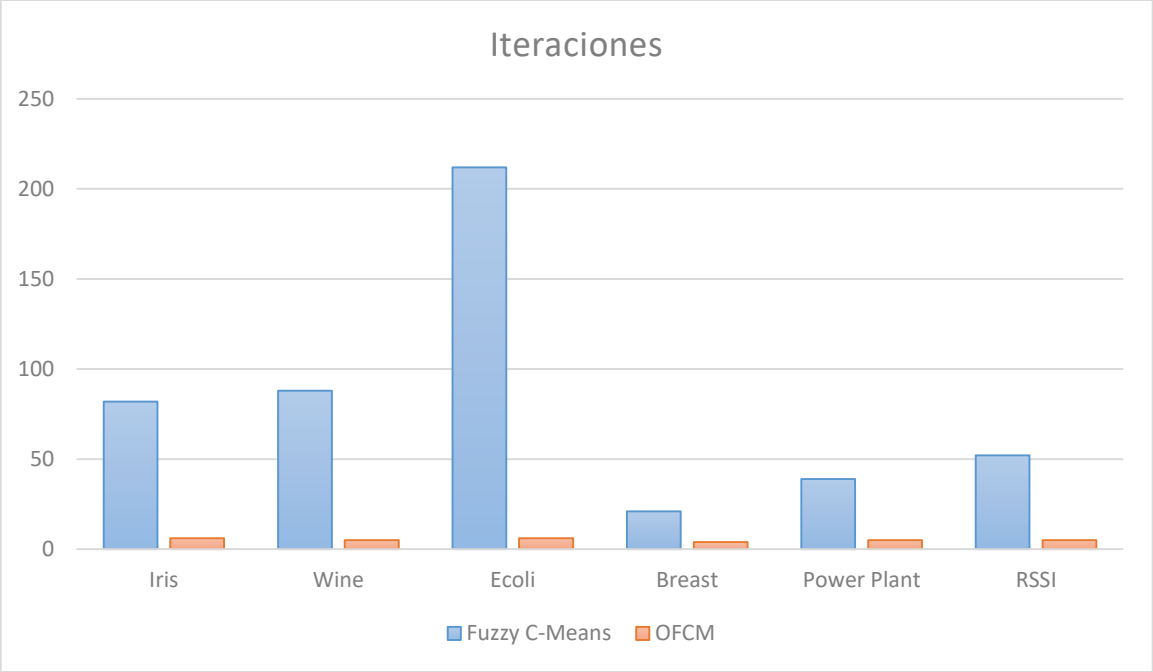

Figura 9. Total de iteraciones entre Fuzzy C-Means y OFCM k=5

Es posible observar que aplicando OFCM, las iteraciones requeridas para procesar las instancias son menor. En cuanto a la convergencia del algoritmo, con base en los análisis realizados, la cantidad de iteraciones realizadas por Fuzzy C-Means es mayor, porque, al aplicar OFCM se recomienda detener el algoritmo con un número considerablemente menor de iteraciones.

Con los resultados de la presente investigación se demuestra que, mediante la aplicación de la heurística OFCM es factible incrementar la eficiencia del algoritmo de agrupamiento Fuzzy C-Means. En el Anexo A se muestran la reducción de los tiempos de ejecución, la diferencia de la calidad y el total de iteraciones con Fuzzy C-Means y con OFCM para las seis instancias utilizadas en la fase de experimentación.

Capítulo 5
Conclusiones y trabajos futuros

El final de una obra debe hacer recordar siempre el comienzo.
Joseph Joubert

Con base en los resultados expuestos en el apartado anterior, en este capítulo se presentan las conclusiones de esta investigación. Asimismo, se proponen nuevos temas para desarrollar futuras investigaciones.

5.1. Conclusiones

Con la presente investigación se desarrolló una mejora al algoritmo Fuzzy C-Means a la que se denomina *Optimizaded Fuzzy C-Means* (OFCM). Esta heurística, consiste en la aplicación de un nuevo criterio de paro en la fase de convergencia del algoritmo, el cual establece que cuando un umbral dado sea menor o igual al porcentaje de objetos que cambian de grupo en una iteración, el algoritmo se detenga.

Los resultados obtenidos mostraron que fue posible reducir el número de iteraciones necesarias para converger, logrando así una disminución en el costo computacional del algoritmo, sin afectar de manera significativa su calidad. Además, se mostró que, mediante la aplicación de la mejora OFCM fue factible incrementar la eficiencia del algoritmo de agrupamiento Fuzzy C-Means.

Cabe destacar que la heurística O-K-Means, que sirvió de base para el desarrollo del nuevo criterio de paro fue desarrollada en el CENIDET para la mejora del algoritmo K-Means y también se demostró que puede ser implementada en otro algoritmo de agrupamiento.

5.2. Trabajos futuros

Para los trabajos futuros se propone:

a) Explorar otras heurísticas propuestas en el CENIDET para aplicarla en este mismo algoritmo de agrupamiento y extender estas mismas heurísticas a otros diferentes algoritmos.
b) Explorar el comportamiento del algoritmo con instancias de gran tamaño para conocer su comportamiento en el campo del *Big Data* y aplicar algunas de las heurísticas desarrolladas en el CENIDET para mejorarlo.
c) Desarrollar una heurística para mejorar el algoritmo Fuzzy C-Means en alguna de sus otras fases.

REFERENCIAS

[1] J. Pérez, C. E. Pires, L. Balby, A. Mexicano, M. A. Hidalgo, "Early Classification: A New Heuristic to Improve the Classification Step of K-Means", *Journal of Information and Data Management*, vol. 4, no. 2, pp. 94-103, June 2013.

[2] J. Pérez, R. Pazos, L. Cruz, G. Reyes, R. Basave, H. Fraire, "Improving the Efficiency and Efficacy of the K-means Clustering Algorithm Through a New Convergence Condition" *Springer-Verlag*, vol. 3, pp. 674-682. 2007.

[3] J. Pérez, M. Hidalgo, N. Almanza, N. Castro, V. López, "Mejora del algoritmo k-means mediante una meta-heurística orientada a la reducción de su complejidad computacional", *Encuentro Nacional de Ciencias de la Computación.* 2014

[4] J. Pérez, A. Mexicano, R. Pazos, R. Santaolaya, M. Hidalgo, A. Moreno, N. Almanza, "Improvement to the K-Means Algorithm Through a Heuristics Based on a Bee Honeycomb Structure", *Journal of Network and Innovative Computing,* vol. 1, pp. 119-125, 2013

[5] J. Pérez, R. Pazos, M. Hidalgo, N. Almanza, O. Díaz-Parra, R. Santaolaya, V. Caballero, "An Improvement to the K-means Algorithm Oriented to Big Data", *Proceedings of the International Conference on Numerical Analysis and Applied Mathematics*, AIP Publishing, vol. 1648. Pp. 820002-1-820002-4. 2014

[6] J. Pérez, A. Martínez, N. Almanza, A. Mexicano, R. Pazos, "Improvement to the K-Means algorithm by using its geometric and cluster neighborhood properties", *Proceedings of ICITSEM*, Dubai, UAE. 2014

[7] J. Pérez, R. Pazos, V. Olivares, M. Hidalgo, J. Ruiz, A. Martínez, N. Almanza, M. González, "Optimization of the K-Means algorithm for the solution of high dimensional instances", *International Conference of Numerical Analysis and Applied Mathematics*, AIP Publishing, pp. 310002-1-310002-5 2016

[8] J. Pérez, N. Almanza, J. Adams, M. González, A. Mexicano, S. Saenz, J.M. Rodríguez, "Improving the Efficiency of the K-medoids clustering Algorithm by Getting Initial Medoids", *Recent Advances in Information Systems and Technologies*, Springer, no. 569. 2017.

[9] N. Almanza, "Desarrollo de heurísticas para la mejora del algoritmo K-Means en las fases de clasificación y convergencia", Tesis de doctorado, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Morelos, México. 2018

[10] A. Vega, “Revisión del Estado del Arte de los Algoritmos K-Means y sus Mejoras”, Tesis de maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Morelos. México. 2017.

[11] D. Mújica, “Segmentación de Imágenes Utilizando Algoritmos Robustos de Agrupamiento Difuso”, Tesis de doctorado, Instituto Politécnico Nacional, México. D.F., México. 2013

[12] R. Suganya, R. Shanthi, “Fuzzy C-Means Algorithm-A Review”, *International Journal of Scientific and Research Publications*, vol. 2, no. 11, Nov 2012.

[13] Gustafson-Kessel Algorithm. R-studio, June 2019. [Online]. Disponible: http://rstudio-pubs-static.s3.amazonaws.com/248394_e80ad79e3f0843ce9631600a59eefcfb.html

[14] Gath, Geva, “Unsupervised Optimal Fuzzy Clustering”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 7, vol. 2, 1989.

[15] Y. Cheng, “Mean Shift, Mode Seeking, and Clustering”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 7, vol. 8, pp. 790-799. 1995.

[16] J. Dunn, “A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters”, *Journal of Cybernectics*, vol. 3, no. 3, pp. 32-57. 1974.

[17] J. Bezdek, “FUZZY C-MEANS: The Fuzzy C-Means Clustering Algorithm”, *Computers & Geosciences*, vol. 10, No. 2-3, pp. 191-203. 1984.

[18] R. Suganya, R. Shanthi, “Fuzzy C-Means Algorithm-A Review”, *International Journal of Scientific and Research Publications*, vol. 2, no. 11, 2012.

[19] F. Kolen, T. Hutcheson, “Reducing the Time Complexity of the Fuzzy C-Means Algorithm”, *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 263-267, 2002.

[20] X. Wang, Y. Wang, L. Wang, “Improving Fuzzy C-Means clustering based on feature-weight learning”, *ELSEVIER Pattern Recognition Letters*, no. 25, pp. 1123-1132. 2004

[21] F. Yu, F. Wu, Q. Sun, “Improving Fuzzy C-Means with Shadow Set”, *School of Mathematical Sciences, Beijing Normal University*. 100875.

[22] J. Zhang, L. Shen, “An Improved Fuzzy C-Means Clustering Algorithm Based on Shadowed Sets and PSO”. *Hindwai Publising Corporation Computational Intelligence and Neuroscience*, vol. 2014, no. 368628. 2014.

[23] C. Lu, S. Xiao, X. Gu, “Improving Fuzzy C-Means clustering algorithm base on a density-induced distance measure”, *The Journal of Engineering,* 2014

[24] A. Stetco, X. Zeng, J. Keane, “Fuzzy C-Means++: Fuzzy C-Means with effective seeding initialization”, *ELSEVIER Expert Systems with Applications*, no. 42, pp. 7541-7548. 2015.

[25] M. Al-Ayyoub, M. Al-Andoli, Y. Jararweh, M. Smadi, B. Gupta, “Improving fuzzy C-mean-based community detection in social networks using dynamic parallelism”, *ELSEVIER Computers and Electrical Engineering*, pp. 1-14. 2018.

[26] C. Merz, P. Murphy, D. Aha, “UCI Repository of Machine Learning Databases. Department of Information and Computer Science”, *University of California*. Jun 2019. [Online] Disponible: http://www.ics.uci.edu/ mlearn/MLRepository

Anexo A

En esta sección se muestran los resultados promedio de la ejecución de todas las instancias utilizadas. En la Tabla A.1, la primera columna indica el nombre de la instancia; la segunda y tercera columna muestran el tiempo de ejecución y el valor de la función objetivo (calidad de la solución) del algoritmo Fuzzy C-Means; y en las dos últimas columnas se observan los resultados obtenidos para OFCM.

Tabla A 1. Resultados de reducción de tiempo y diferencia de la calidad de solución con Fuzzy C-Means y OFCM

Instancia	**Fuzzy C-Means**		***Optimizaded* Fuzzy C-Means**	
	z^*	**Tiempo**	z^*	**Tiempo**
Iris	166.75	13.34	166.99	2.54
Wine	73730437.5	128.81	74162920.1	42.67
Ecoli	44.24	59.19	45.35	9.27
Breast	29302.9	11.62	30487.7	3.85
Power Plant	4827072.1	618.6	4993670.1	136.35
RSSI	20593891.9	403.37	20985467.5	77.3

Mediante el análisis de los resultados presentados en la Tabla anterior, se observan altos porcentajes de ahorro de tiempo en todas las instancias. En los experimentos, para cada instancia se utilizó un valor diferente del umbral U de acuerdo al resultado obtenido en la aplicación del Principio de Pareto.

En las Figuras mostradas a continuación, se muestra la diferencia de tiempos de ejecución y el total de iteraciones de manera comparativa entre el algoritmo Fuzzy C-Means y OFCM, con una corrida de cada una de las instancias.

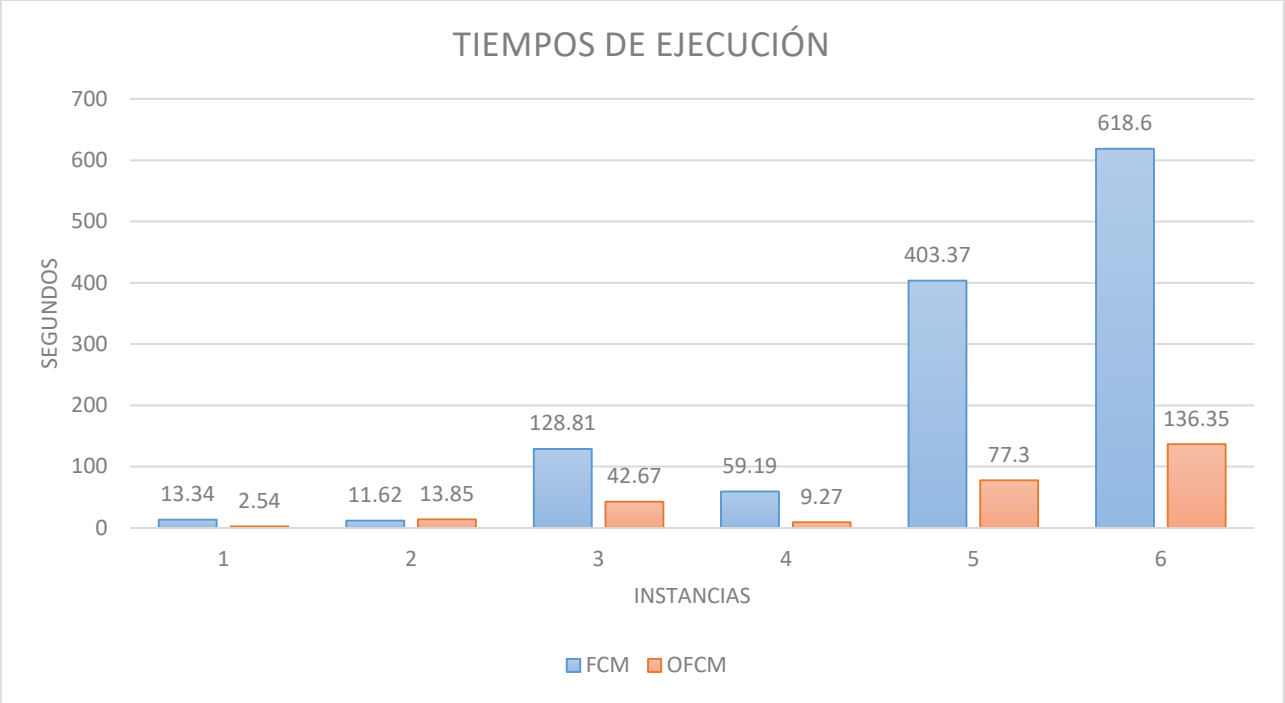

Figura A. 1 Diferencia de tiempo de ejecución entre Fuzzy C-Means y OFCM

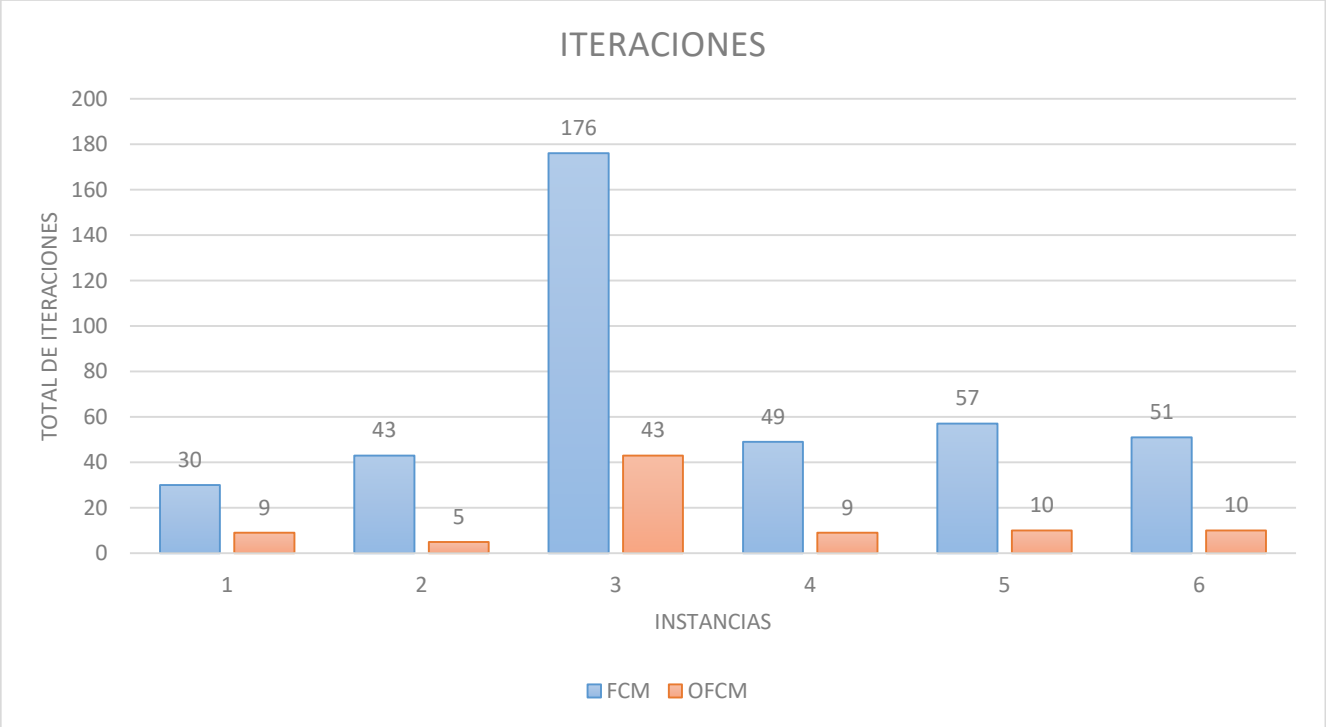

Figura A. 2 Total de iteraciones de Fuzzy C-Means y OFCM

Es posible observar que aplicando OFCM, el tiempo requerido para procesar las instancias es menor, en cuanto a la convergencia del algoritmo, la cantidad de iteraciones realizadas por Fuzzy C-Means es mayor, ya que, con base en los análisis realizados, OFCM recomienda detener el algoritmo con un número mucho menor de iteraciones.